Informatyka - Studio Projektowe 1 - City Police Simulation

Documentation

Bartłomiej Kręgielewski, Szymon Lepianka, Małgorzata Pinior

1. Opis

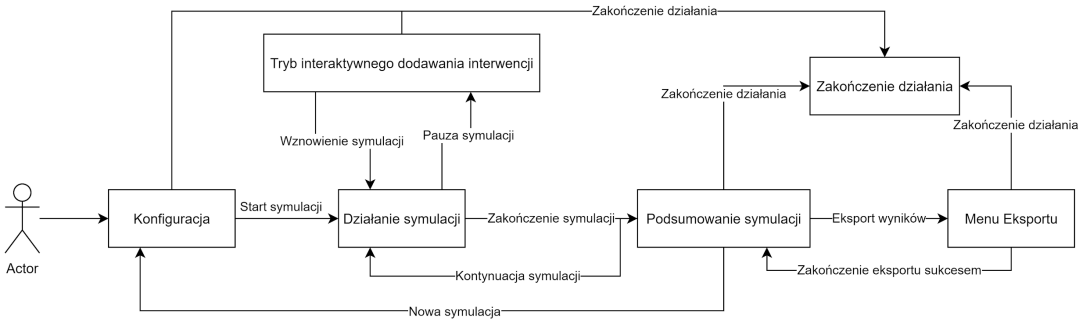
Aplikacja ma na celu symulację pracy jednostek policji w dowolnie wybranym mieście. Możliwe jest wybranie dodatkowych parametrów symulacji, tak aby jak najbardziej przybliżyć logikę i działanie policji w każdym mieście.

2. Wymagania funkcjonalne

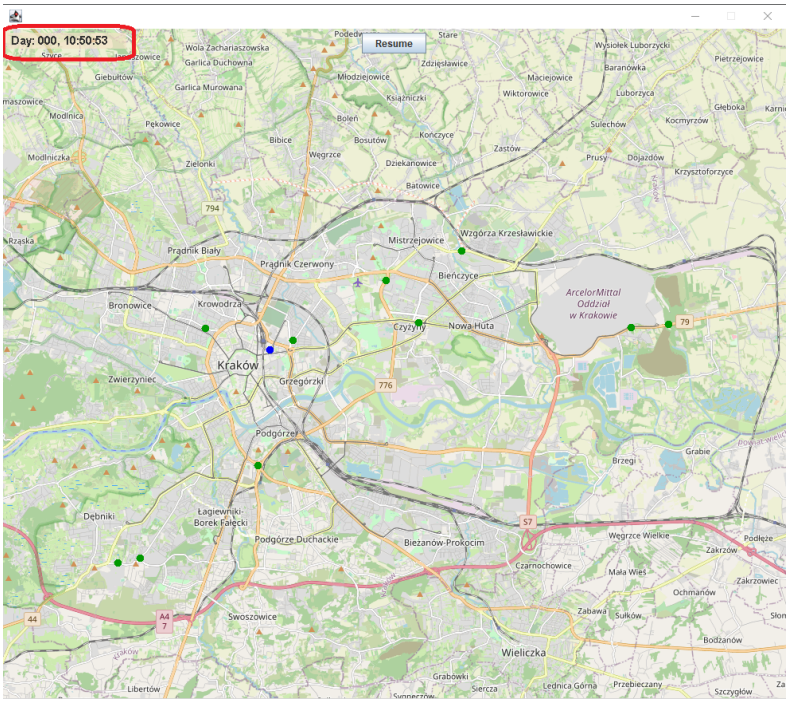
a. Symulacja patroli i interwencji
   i. Zasady działania patrolu opisane w punkcje 5.
   ii. Cykl dnia i nocy (wpływa na szybkość przemieszczania patroli, ilość interwencji)
b. Moduł zarządcy odpowiedzialny za generowanie losowości wybranych zdarzeń
c. Moduł odpowiedzialny za zarządzanie patrolami - nazwany HQ. Odpowiedzialny za przydzielanie zadań i rozsyłanie dyspozycji do patroli
d. Monitoring i eksport danych z symulacji
   i. Eksport rezultatów do pliku CSV
   ii. Logi symulacji
e. Integracja z mapami pozwalająca na dynamiczne generowanie rzeczywistego świata symulacji na bazie dostępnych map
f. Interfejs użytkownika
   i. Konfiguracja początkowych parametrów symulacji
   ii. Wizualizacja przebiegu symulacji
   iii. Możliwość pauzy symulacji
      1. Możliwość interakcji np. Poprzez umieszczenie interwencji w danym punkcie, dodanie jednostki policji
      2. Możliwość dodania lub usunięcia wybranych interwencji

3. Workflow

- Diagram prezentujący działanie aplikacji



- Dostępne ustawienia konfiguracji:
  - Lokalizacja - można wybrać odpowiednio:
    - Kraj
    - Miasto
  - Poziom zagrożenia dla każdej dzielnicy osobno:
    - Skala trzystopniowa dotycząca poziomu zagrożenia dzielnicy
    - Maksymalna ilość interwencji na godzinę - możliwość wyboru prawdopodobieństwa wystąpienia interwencji, zależnie od poziomu zagrożenia w danej dzielnicy
    - Prawdopodobieństwo, że interwencja przerodzi się w strzelaninę - ustawiana osobno dla każdego poziomu bezpieczeństwa dzielnicy
  - Aktywny posterunek (HQ) - możliwość wyboru na mapie danego miasta wyboru miejsca, gdzie ma znajdować się posterunek policji, od którego wszystkie patrole zaczną swoje działanie
  - Ilość jednostek policji na całe miasto:
    - Możliwość wyboru ilości jednostek policji na jedną zmianę (w symulacji stosujemy tryb pracy trzyzmianowy)
  - Simulation Time Rate - przelicznik, ile razy szybciej symulacja działa w stosunku do czasu rzeczywistego
  - Czas trwania symulacji - czas, ile ma trwać nasza symulacja (po wpisaniu na przykład 60 sekund, symulacja zakończy działanie po minucie)



Zdjęcie prezentuje wygląd mapy z zaznaczonym okienkiem zegara.

4. Opis architektury

a. Opis “świata”

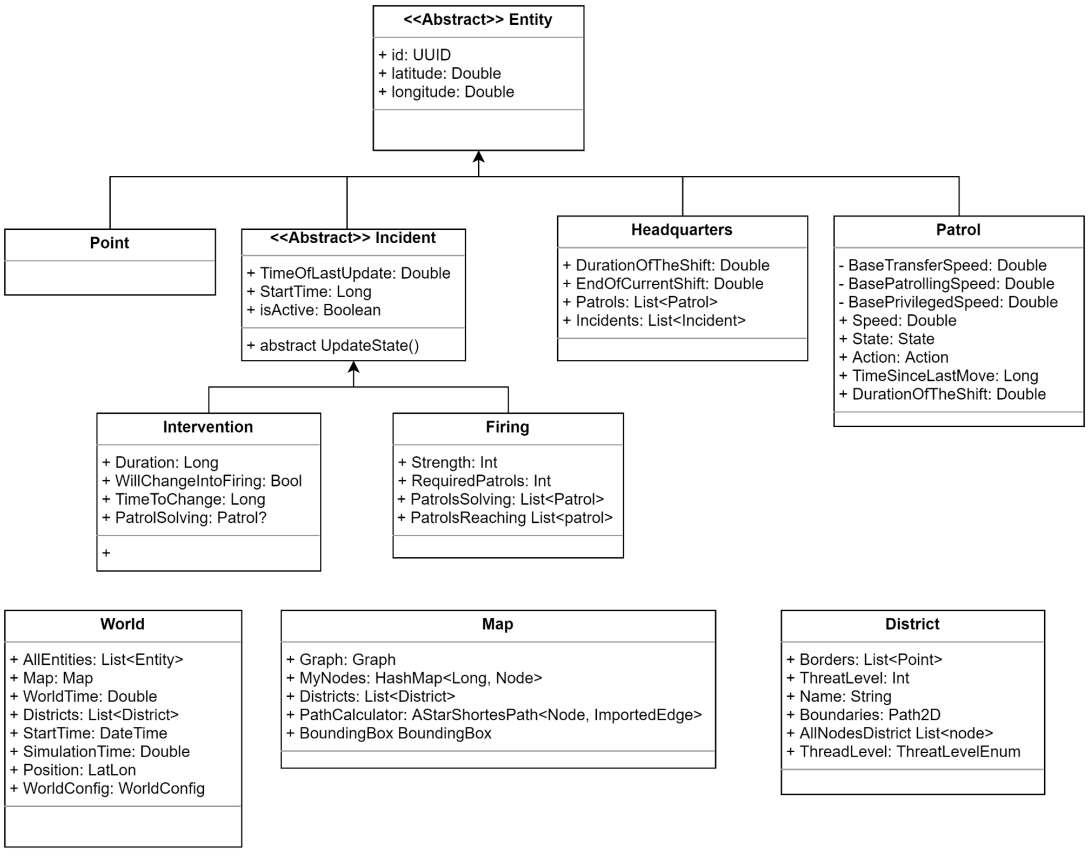


Diagram prezentuje z jakich elementów składa się świat wewnątrz symulacji

b. Entities - jednostki w symulacji, z których składa się większość obiektów w symulacji. Można je podzielić na trzy główne obszary:
    i. Patrol
    Każdy agent, czyli Patrol, jest jednostką w symulacji. Możliwe stany i akcje, jakie wykonuje są opisane w punkcie Opis Agentów.
    ii. HQ - Headquarters
    Punkt centralny symulacji, w którym agenci zaczynają swoje patrole i wracają gdy kończą zmianę. HQ zajmuje się przydzielaniem zadań i wysyłaniem patroli na interwencje bądź też strzelaninę.
    iii. Graph
    Świat jest reprezentowany przez graf, gdzie węzły są połączone krawędziami. Każda krawędź (Edge) reprezentuje możliwe przejście między węzłami (drogę), gdzie drogi dwukierunkowe są reprezentowane przez dwie krawędzie, a jednokierunkowe przez jedną. Jednakże w symulacji dla uproszczenia uznajemy, że wszystkie drogi są dwukierunkowe. Więcej informacji w punkcie Działanie map.
c. Logger
    i. Zapisywanie zdarzeń do pliku jak i wyświetlanie bezpośrednio do użytkownika poprzez GUI.
    ii. Logger zapisuje do pliku w formacie *data_godzina.log*, gdzie data i godzina to czas włączenia symulacji.
    iii. W pliku znajdują się informacje w formacie:
    Data czas_symualcji informacja
    Gdzie data to czas wystąpienia zdarzenia, czas w symulacji to liczba sekund od rozpoczęcia symulacji.
    iv. Na koniec symulacji jest podsumowanie najważniejszych akcji, jakie wydarzyły się w trakcie symulacji.

d. Wątki
    i. Police_AI
    Główna pętla symulacji, która składa się z czterech zadań. Najpierw HQ przydziela każdej jednostce zadanie. Następnie aktualizuje się status agentów. Po tym każda jednostka wykonuje przydzielone akcje zależne od stanu, w którym obecnie się znajduje. Zadanie sleep() ma na celu spowolnienie wykonywania obliczeń w symulacji.

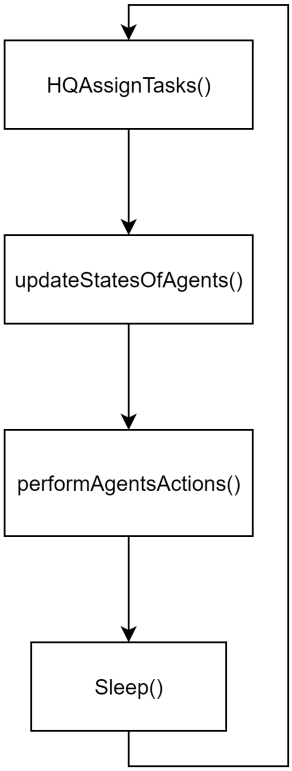


Diagram prezentuje główną pętlę działania programu.

ii. UI Thread
Wątek odpowiedzialny za wyświetlanie GUI i rysowanie jednostek na mapie.

iii. Events Director
Jest to moduł odpowiedzialny za losowanie zdarzeń i parametrów danych zdarzeń. Raz na godzinę usuwa z mapy patrole w stanie “neutralized”.

iv. Events Updater
Jest odpowiedzialny za aktualizowanie zdarzeń. W przypadku zwykłej interwencji, jeśli jej czas trwania (czas potrzebny przez patrol do jej rozwiązania) minął, zmienia interwencję na nieaktywną. W przypadku interwencji, która zmienia się w strzelaninę, zmienia stan danej interwencji w strzelaninę. W przypadku strzelaniny odpowiednio obniża poziom ‘strength’ w zależności od ilości patroli uczestniczących. Zmienna ‘strength’ jest losowana dla każdej strzelaniny. Można ją utożsamiać z tym, ile patroli i przez jaki czas muszą rozwiązywać daną strzelaninę. W przypadku gdy wartość ‘strength’ zostanie zmniejszona do 0, strzelanina zostaje zakończona. Parametr “strength” to inaczej “siła” danej strzelaniny, czyli jak długo patrolom zajmie jej zakończenie.

5. Opis agentów

a. State machine

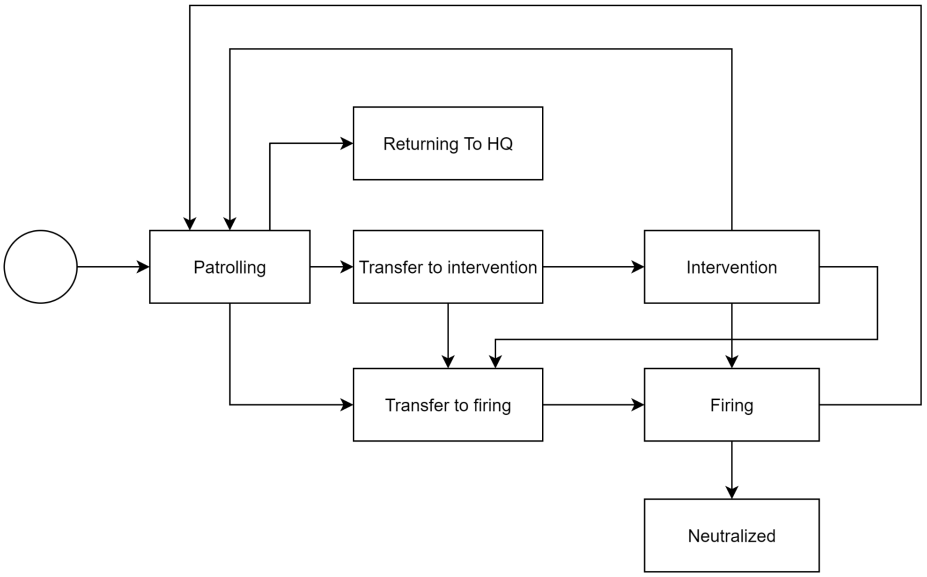
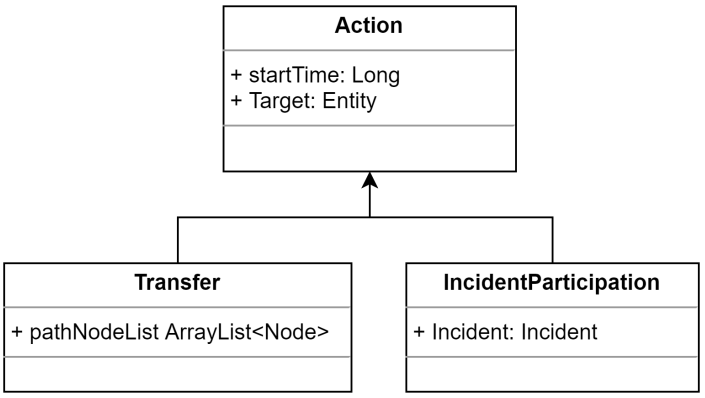


Diagram prezentujący maszynę możliwych stanów, w których znajduje się jednostka policji

Patrol może znajdować się w 7 różnych stanach. Podstawowym stanem jest ”patrolling”, które może zamienić się w “transfer to intervention”, “transfer to firing” lub “returning to HQ” po skończonej zmianie. Do tego dochodzą stany “intervention”, która może przerodzić się w strzelaninę, a więc dodatkowy stan “firing”. Podczas strzelaniny jednostki mogą przejść w stan “neutralized”, czyli patrol został zabity lub postrzelony, a więc musi skończyć służbę (nie może wrócić do stanu “patrolling”).

b. Możliwe akcje, czyli działania jakie może podjąć patrol.



Graf prezentuje podział akcji na transfer i uczestnictwo w incydencie

W symulacji wyróżniamy dwie możliwe akcje (działania). Jest to transfer, a więc przejazd patrolu na strzelaninę, interwencję, patrolowanie okolicy lub powrót do HQ. Drugą akcją jest udział w incydencie, do których zaliczany interwencję i strzelaninę. Patrol w symulacji może zawsze wykonywać tylko

jedną z tych dwóch akcji. Klasa Action została zdefiniowana, aby mieć obiekt, w którym zapiszemy wszystkie najważniejsze informacje, jakich potrzebuje jednostka wykonująca jedną z tych dwóch akcji.

c. Praca jednostek
Patrole pracują w systemie 3 zmianowym - co 8 godzin następuje zmiana patroli. Patrole pojawiają się w HQ i mogą albo patrolować, albo jechać na interwencje. W momencie końca zmiany, z HQ wyjeżdżają patrole nowej zmiany (ilość ustalona przed rozpoczęciem symulacji). Po zakończeniu zmiany, jeżeli są w stanie “patrolling”, wracają do HQ, po czym kończą działanie.
Patrol posiada 3 różne prędkości poruszania się, zależnie od akcji, jaką wykonuje.
    - Podczas patrolowania prędkość poruszania wynosi 40 km/h.
    - Podczas transferu na interwencję prędkość poruszania wynosi 60 km/h.
    - Podczas transferu na strzelaninę prędkość poruszania wynosi 80 km/h.

d. Jednostki policji i możliwe akcje
Na mapie można wyróżnić kolorami wszystkie jednostki policji, określić ich aktualny stan oraz zobaczyć, w którym miejscu nastąpi jedna z dwóch możliwych akcji - interwencja lub strzelanina.
    i. **Ciemnoniebieski** - HQ, posterunek policji wybrany przez użytkownika przed rozpoczęciem symulacji.
    ii. **Zielony** - jednostka w stanie ‘patrolling’, patroluje okolice.
    iii. **Ciemnozielony** - jednostka wracająca do HQ, po skończeniu zmiany, znajduje się w stanie “returning to HQ”.
    iv. **Pomarańczowy** - jednostka policji skierowana na interwencję, znajduje się w stanie “transfer to intervention”.
    v. **Czerwonawy** - jednostka policji skierowana do strzelaniny, znajduje się w stanie “transfer to firing”.
    vi. **Niebieski** - jednostka policji uczestnicząca w interwencji, znajduje się w stanie “intervention”. Aktualnie interwencja (punkt koloru czerwonego) jest zakrywa przez jednostkę (kolor niebieski), która rozwiązuje daną interwencję.
    vii. **Fioletowy** - jednostka policji uczestnicząca w strzelaninie, znajduje się w stanie “firing”. Aktualnie strzelanina (punkt koloru czarnego) jest zakrywa przez jednostki (kolor fioletowy), które rozwiązują daną strzelaninę.
    viii. **Biały** - jednostka policji zneutralizowana, znajduje się w stanie “neutralized”.
    ix. **Różowy** - jednostka policji obliczająca trasę do celu, znajduje się w stanie “patrolling”, ale na ten moment wybiera trasę - dzieje się to w osobnym wątku, aby uniknąć możliwości zacięcia symulacji.
    x. **Czerwony** - miejsce interwencji, nieprzerodzonej w strzelaninę.
    xi. **Czarny -** miejsce strzelaniny.

6. Opis zasad wydarzeń w symulacji

W symulacji wyróżniamy trzy główne zdarzenia i ich zasady, według których zachowują się agencji, czyli Patrole, w poszczególnych sytuacjach:

a. Interwencja
   i. Losowanie miejsca - odbywa się w klasie IncidentFactory
   ii. Losowanie czasu trwania interwencji - odbywa się w klasie IncidentFactory
   iii. Tylko jeden patrol może zostać skierowany na daną interwencję, jest wybierany najbliższy patrol znajdujący się w okolicy o zadanym promieniu. Jeśli nie znaleziono patrolu, promień się powiększany, aż do znalezienia najbliższej jednostki.
   iv. Jak interwencja przerodzi się w strzelaninę:
      1. Ustalany jest poziom prawdopodobieństwa przed rozpoczęciem symulacji, że interwencja przerodzi się w strzelaninę.
      2. Czas trwania strzelaniny przedstawiony jest jako “strength”. Dokładny opis znajduje się w punkcie “strzelanina”.
      3. Losowane parametry opierają się na lokalizacji (poziom bezpieczeństwa dzielnicy) oraz porze dnia. Zła dzielnica zwiększa prawdopodobieństwo strzelaniny, tak samo jak późna pora dnia. Możliwe poziomy bezpieczeństwa dzielnicy:
         a. Safe
         b. RatherSafe
         c. NotSafe

   v. Przerwanie dojazdu na interwencję może nastąpić tylko w przypadku wezwania jako wsparcie do strzelaniny (patrol, który jest w trakcie interwencji nie może zostać z niej wezwany). Wtedy występują następujące akcje:
      1. Interwencja jest niewykonana - wraca jako zadanie do HQ, które wymaga przydzielenia innej jednostki
      2. Patrol przydzielony jako wsparcie do strzelaniny:
         a. Może zostać odwołany, jeśli na miejsce strzelaniny dostarczono wymaganą liczbę wsparcia, wtedy Patrol przechodzi w stan patrolowania - ponieważ wcześniejsze zadanie z interwencją wróciło do HQ i wymaga ponownego przydzielenia patrolu

3. Dotarcie do strzelaniny oznacza zmianę stanu patrolu na “firing”

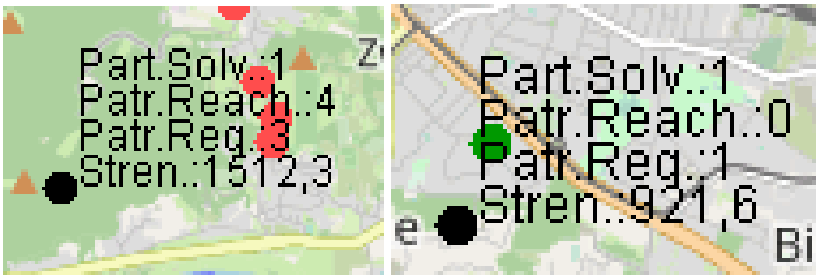
vi. Czas dojazdu na interwencję:

1. Może być większy niż czas obliczony według prędkości patrolu i odległości przez sytuację drogową (światła lub korek) - mowa tutaj o interwencji nie wymagającej używania sygnałów dźwiękowych i świetlnych.

b. Strzelanina

i. Pojawia się tylko w przypadku regularnej interwencji (przeradza się z interwencji)

ii. Czas trwania strzelaniny przedstawiony jest jako “strength”. Wartość “strength” losowana jest dla każdej strzelaniny przed jej rozpoczęciem. Jednostki skierowane do niej, mogą zmniejszyć wartość “strength” - jest to funkcja liniowa, czym więcej jednostek, tym szybciej można zakończyć daną strzelaninę.



Grafiki prezentują miejsce strzelaniny wraz ze statystykami.

iii. Ilość potrzebnego wsparcia do strzelaniny wybierana losowo - za przydział odpowiada HQ. Sugeruje się wybór większej liczby niż potrzeba. Zasady działa doboru jednostek wsparcia:

1. HQ wybiera jednostki w stanie obserwacji z określonego promienia wokół strzelaniny. Jeśli ilość jednostek jest niewystarczająca, promień się powiększa i wybierane są najbliższe jednostki w stanie obserwacji
2. Dalszy brak jednostek skutkuje wybraniem Patroli, które są w trakcie jechania na interwencję
3. Jeśli dalej brakuje jednostek, HQ czeka aż patrole w zadanym promieniu przejdą ze stanu interwencji do stanu obserwacji
4. Nadmiarowe jednostki patrolu zostają odwołane i przechodzą w stan obserwacji

iv. Ucierpienie jednostki podczas interwencji

1. Prawdopodobieństwo postrzelenia lub śmierci Patrolu wynosi 1% na godzinę. Czyli co godzinę dla każdej jednostki losuje się z prawdopodobieństwem 1%, czy jednostka zostanie ranna w tej godzinie.
2. Ranna jednostka nie przechodzi do stanu “patrolling”, tylko zostaje zneutralizowana, jej pozycja jest zaznaczona na mapie, dopóki process Events Director jej nie usunie, co następuje co pełną godzinę czasu symulacji.

v. Czas dojazdu na strzelaninę:

1. Może być mniejszy niż czas obliczony według prędkości patrolu i odległości przez przejazd uprzywilejowany - wymagane użycie sygnałów dźwiękowych i świetlnych.

c. Patrol obserwujący
   i. Porusza się po okolicy, zazwyczaj powoli, czasem się zatrzymuje (w symulacji na ułamek sekundy, kiedy wybiera następny cel patrolowania).
   ii. Może zostać wezwany do interwencji, strzelaniny lub po 8 godzinach kończy zmianę i wraca do HQ.

7. Wykorzystywane technologie

a. Pobranie informacji o danym mieście:
   Do wyboru mieliśmy OpenRouteStreet, GoogleMaps oraz OpenStreetMap. GoogleMaps wymaga płatnej subskrypcji, aby móc pobierać informacje o danym mieście, dlatego z tego powodu zostało odrzucone (subskrypcja przekraczała znacznie fundusze studentów). OpenRouteStreet natomiast nie pozwalał pobierać danych z map, a tylko dokładną trasę między punktami. Dla działania aplikacji potrzebujemy danych na temat miasta, aby móc ją zaprezentować w aplikacji. Z tego powodu wybraliśmy OpenStreetMap, który posiada API. Dzięki temu możemy korzystać z mapy i selekcjonować, co konkretnie z mapy potrzebujemy.
   i. https://wiki.openstreetmap.org/wiki/API
b. Jak w punkcie wyżej, aby móc usprawnić obsługę map, korzystamy z Overpass API, aby móc wygodnie pobierać informacje o danym mieście bez konieczności pobierania wszystkich danych. Dzięki temu mamy możliwość wybrania w symulacji dowolnego miasta, bez konieczności dużych zmian w kodzie. API korzystające z OSM:
   i. Overpass API
c. Po pobraniu interesujących nas danych, potrzebowaliśmy czegoś, co zamieni te dane na graf, który byłby naszą reprezentacją mapy. Szybko udało się znaleźć OSM to JGraphT:
   i. OsmAPI
d. Jak w punkcie wyżej, dzięki znalezieniu odpowiedniej biblioteki do przerabiania danych z OpenStreetMap na graf, zdecydowaliśmy się użyć tej samej biblioteki do grafów, aby móc z nich korzystać i wykonywać na nich odpowiednie operacje. Biblioteka do grafów:
   i. JGraphT
e. Aby wizualizacja była przyjemna dla oka, wybraliśmy bibliotekę do wizualizacji map, która bazuje na bibliotece SWING, z której korzystamy do tworzenia GUI. Wcześniej rozważany był component SWINGA JXMapViewer, jednak nie jest to rozwijana biblioteka, natomiast JXMapViewer2 jest kontynuacją poprzedniej i działa dla każdej wersji Swinga.
   i. msteiger/jxmapviewer2: JXMapViewer2 (github.com)

f. Do przetwarzania wyników do pliku CSV użyjemy OpenCSV, jest to prosta w użyciu biblioteka do tworzenia i zapisywania plików .csv:
    i. OpenCSV
g. Java 15
    i. Do tworzenia GUI wybraliśmy bibliotekę SWING, ponieważ na niej oparte są poprzednie biblioteki umożliwiające wizualizację grafu i mapy.

8. Działanie map

W aplikacji jest możliwość wybrania dowolnego miasta, działa to na zasadzie wysłania zapytania do API openstreetmap Overpass API. Uzyskujemy wtedy informacje o danym mieście - ulice, rodzaje ulic (jedno-, dwukierunkowa). Po wykonaniu zapytania te informacje zapisywane są do pliku, który jest odczytywany i przetwarzany na graf z użyciem biblioteki OsmAPI. Jeśli już wcześniej użyto pliku .osm z identycznymi informacjami o danym mieście (w formacie XML), jaki zwraca OpenStreetMap API, to przetworzenie tych “surowych” danych na graf zajmuje ok. 2s. Natomiast, jeśli jest to nowa mapa, która wymaga pobrania informacji z API, a następnie ich przetworzenia, cały proces zajmie ok. 20s (dla Krakowa - dla innych miast w zależności od wielkości miasta i prędkości łącza internetowego).

Do obliczania długości tras używamy algorytmu A*, który dostępny jest w bibliotece grafów (JGraphT).

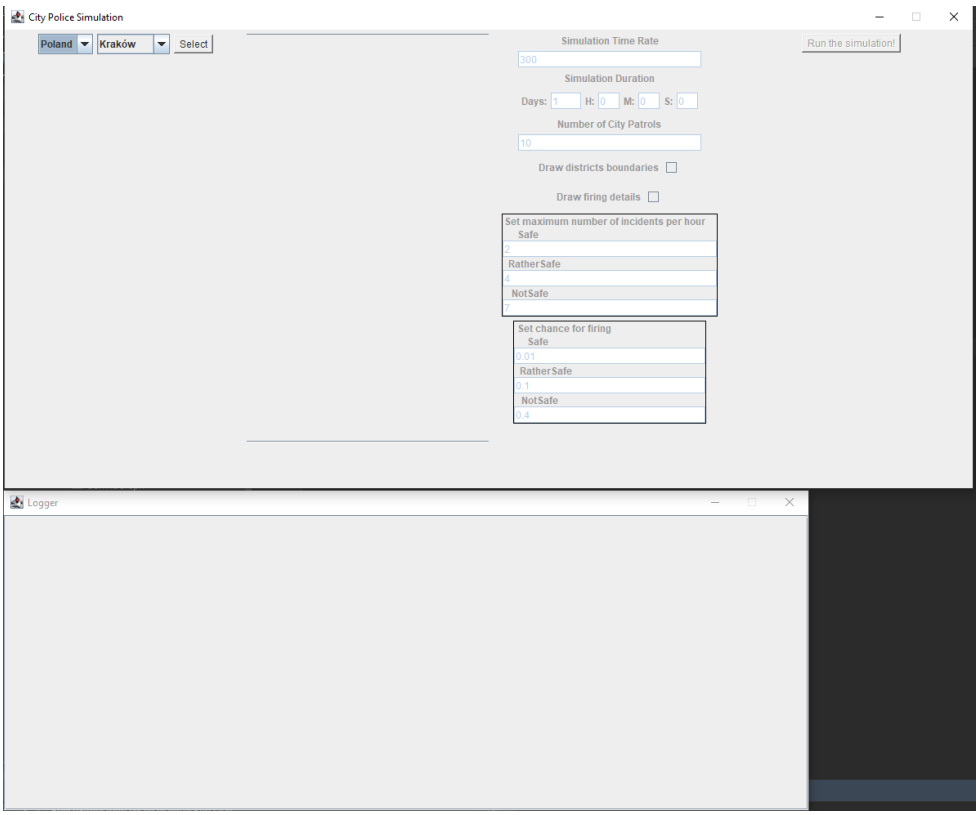
Aby ustalić konkretną lokalizację na mapie, na przykład posterunku (HQ), którą wybiera użytkownik, używa się funkcji, która znajduje najbliższy “Node” na mapie dla dowolnego punktu. Dzięki temu, jesteśmy w stanie obliczać trasę między dowolnymi punktami.

Wydobycie informacji o dzielnicach:

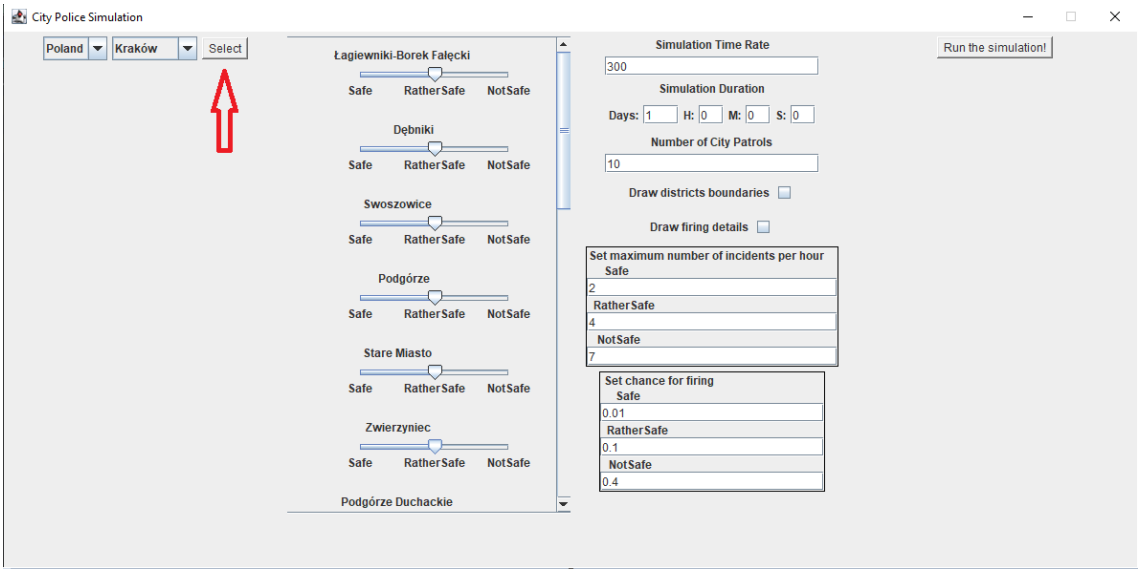
a. Overpass api - możliwość wykonania requestu, aby otrzymać plik z danymi o dzielnicach, podobnie jak w przypadku danych o mieście
b. Zbudowanie grafu zamkniętego, składającego się z węzłów zwróconych w zapytaniu do API. Odbywa się to na zasadzie segregowania węzłów i łączenia ich linią, aby stworzyć zamknięty obszar i wyrysować go na mapie

9. Wygląd aplikacji

W poniższym punkcie przedstawimy jak krok po kroku korzystać z aplikacji wraz z pokazanymi przykładami.
Pierwsze dwa ekrany, które pokażą się po włączeniu aplikacji to panel loggera, na którym będą wyświetlane informacje w trakcie symulacji oraz panel konfiguracyjny - główny panel, na którym będzie prezentowana symulacja.

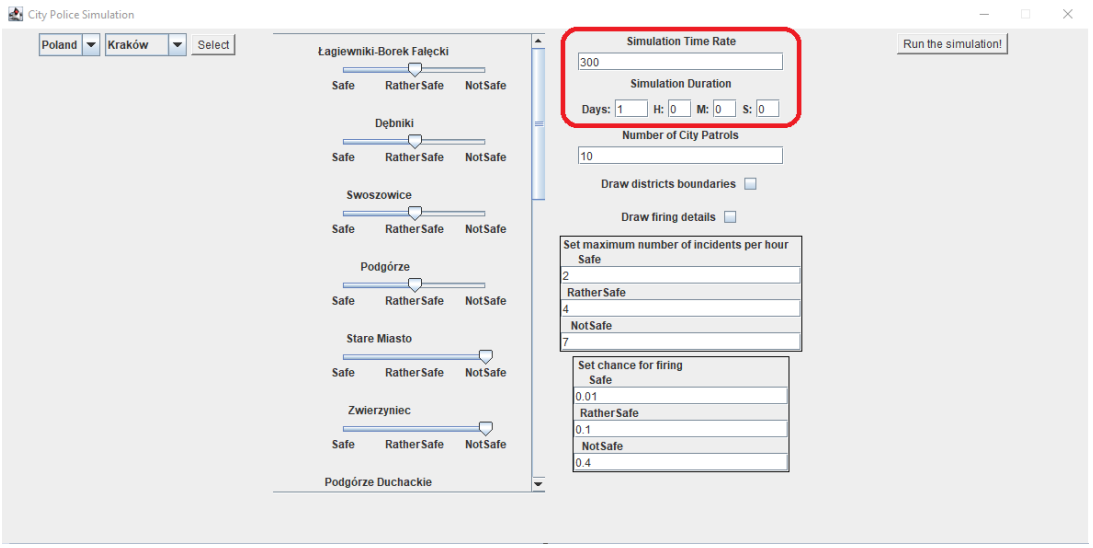


Aby móc odblokować dalszą część konfiguracji, należy wybrać kraj oraz miasto i nacisnąć przycisk “Select”. To spowoduje załadowanie się po chwili okna z wszystkimi dzielnicami danego miasta. Tutaj następuje wybór poziomu bezpieczeństwa dla każdej dzielnicy. Suwakiem należy wybrać wartość, dla każdej dzielnicy, domyślnie ustawiona jest opcja “Rather Safe”. Te czynności są zapisywane w loggerze.

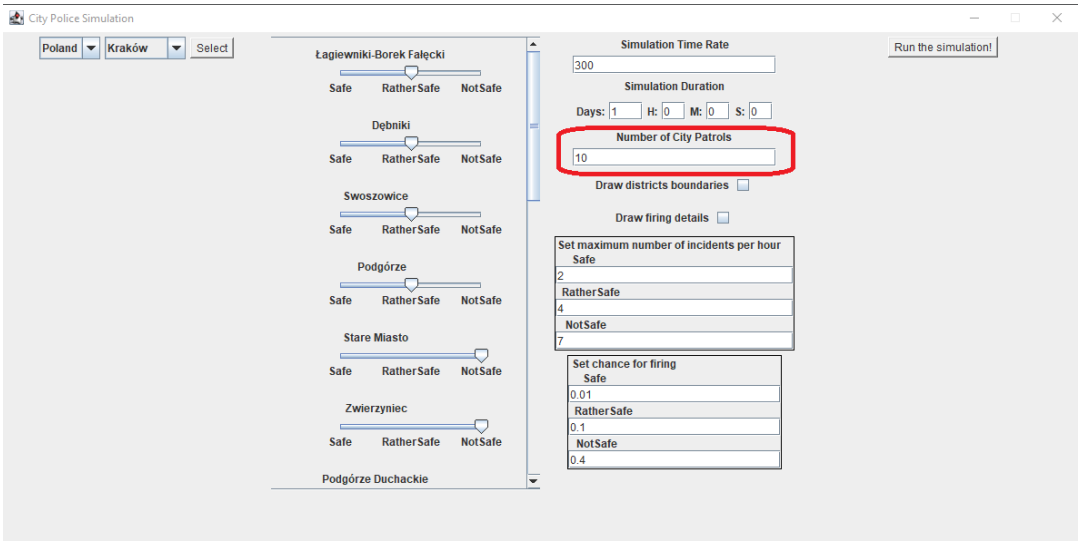


Wszystkie przedstawione parametry mają swoją domyślną wartość. Użytkownik natomiast może je zmieniać według własnego uznania.
Opcja “Simulation Time Rate” oznacza, jak bardzo chcemy “przyspieszyć” symulację. Nie zaleca się stosowania wartości powyżej 800. Przyspieszenie jest na tyle szybkie,

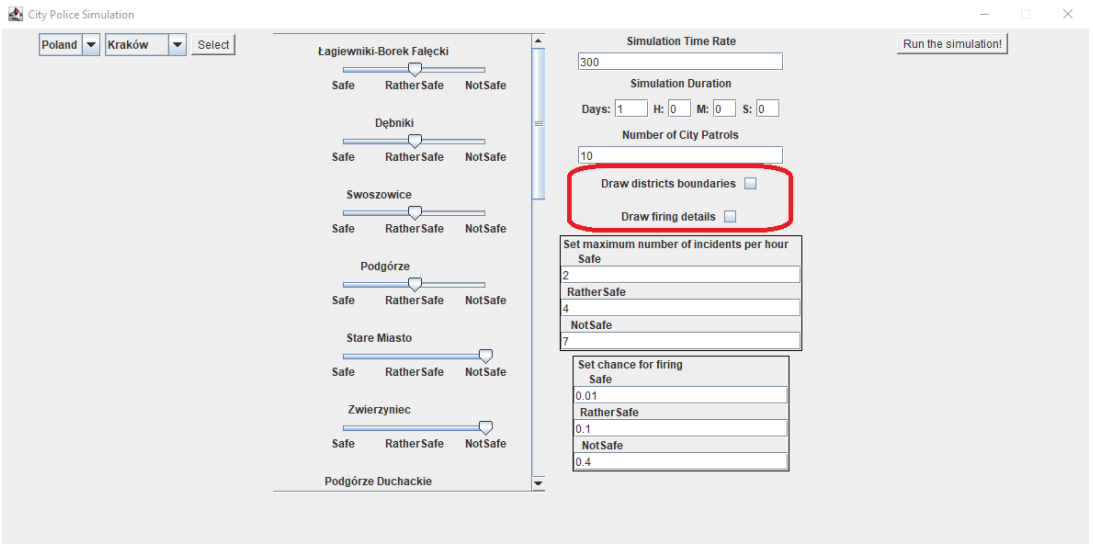
że ciężko nadążyć za wizualizacją poruszających się patroli oraz mogą dawać złudzenie “skakania” po mapie.
Następnie można wybrać czas, jaki chcemy zasymulować. Domyślnie jest to jeden dzień.



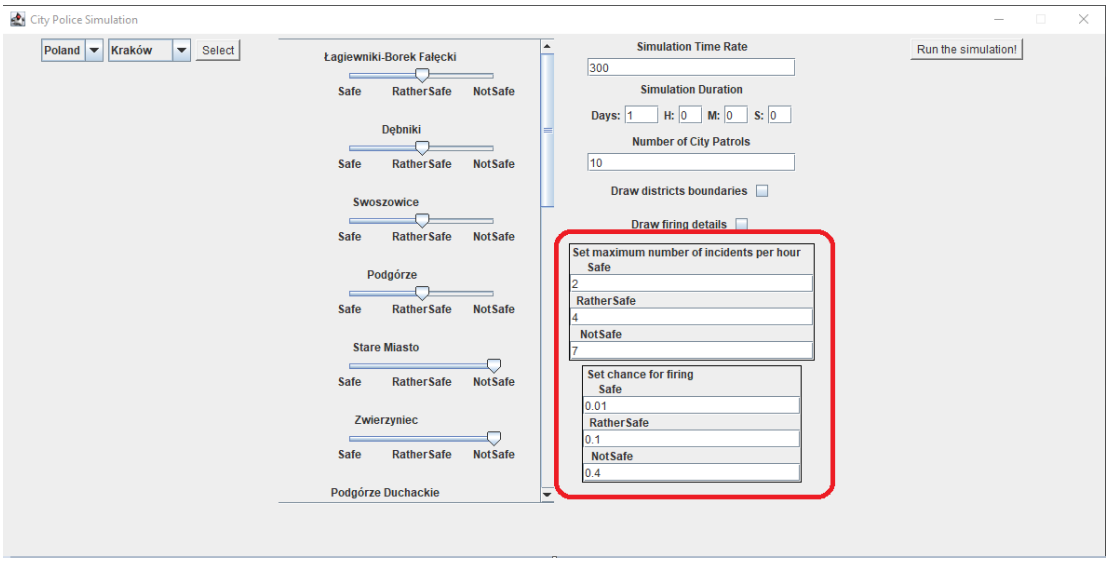
W następnym okienku można wybrać ilość patroli policji na całe miasto. Zalecana wartość nie powinna przekraczać 130. Przy wartości powyżej 80 patroli należy zmniejszyć Time Rate, aby symulacja była bardziej płynna.



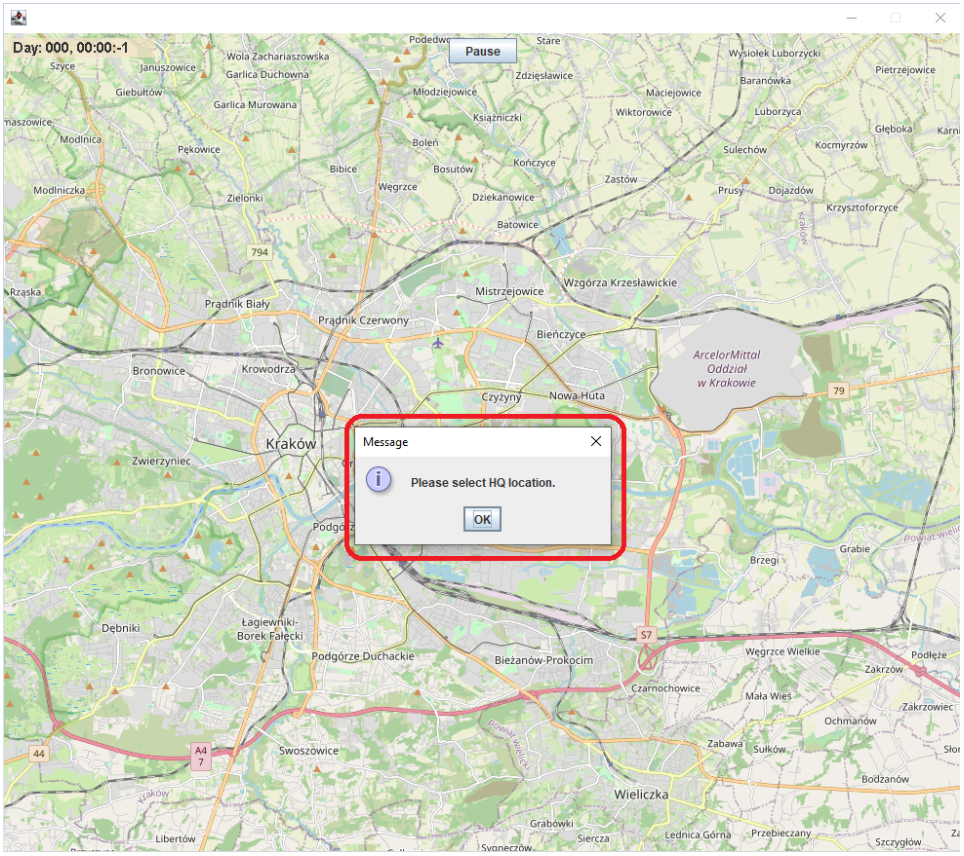
Następne dwie opcje dają możliwość wizualizacji granic dzielnic oraz wyświetlenie statystyki podczas wystąpienia strzelaniny. Statystyki składają się z ilość patroli, które rozwiązują daną strzelaninę, ilości patroli wymaganych do danej strzelaniny, ilości patroli, które są w zasięgu i mogą zostać skierowane do strzelaniny oraz “strength”, czyli długości danej strzelaniny.



Ostatnie dwa okienka przed włączeniem symulacji opisują ile incydentów (przez incydent rozumiemy interwencję i strzelaninę, pamiętając, że strzelanina tworzy się losowo z interwencji) w ciągu godziny ma się pojawić w dzielnicy odpowiednio dla poziomu Safe, RatherSafe oraz NotSafe.
W drugim okienku mamy możliwość ustawienia prawdopodobieństwa wystąpienia strzelaniny.

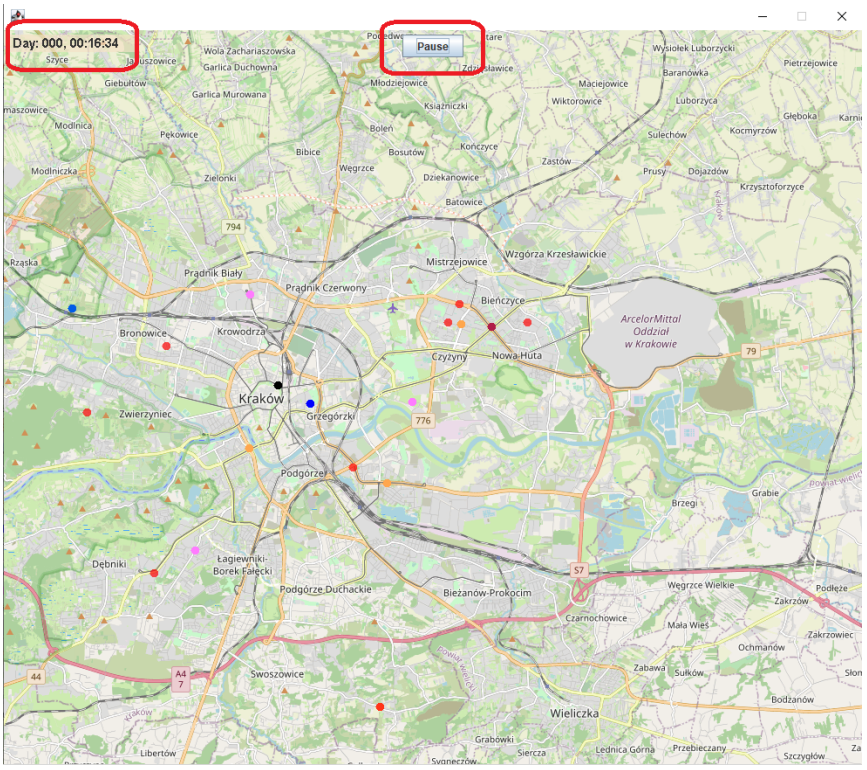


Kiedy wszystkie parametry są dobrane, należy kliknąć przycisk “Run the simulation!”. W okienku pokaże się mapa miasta, które zostało wybrane. Należy wybrać lokalizację głównego posterunku policji, przypomni o tym za każdym razem wyskakujące okienko informacyjne.

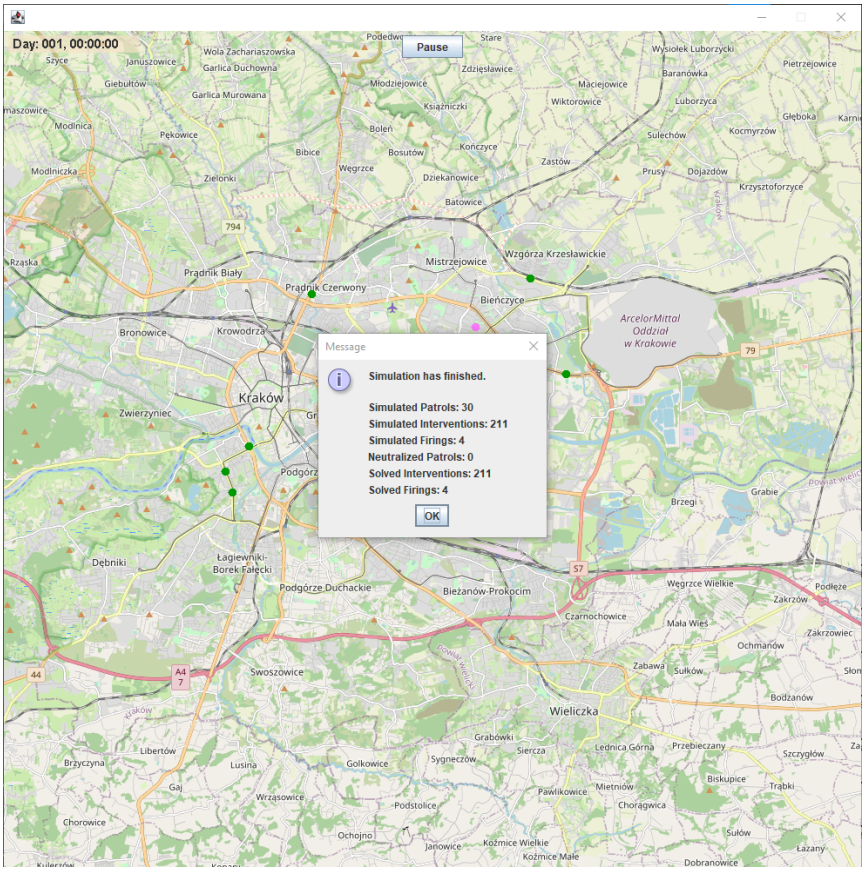


Aby wybrać miejsce HQ należy kliknąć w wybrane miejsce raz, pojawi się niebieska kropka informująca, że jest to główny posterunek. Symulacja rozpoczyna się zaraz po wybraniu miejsca.

Na samej górze ekranu mamy licznik, mówiący o tym, ile czasu minęło już w symulacji oraz przycisk “pause” pozwalający zatrzymać symulację.

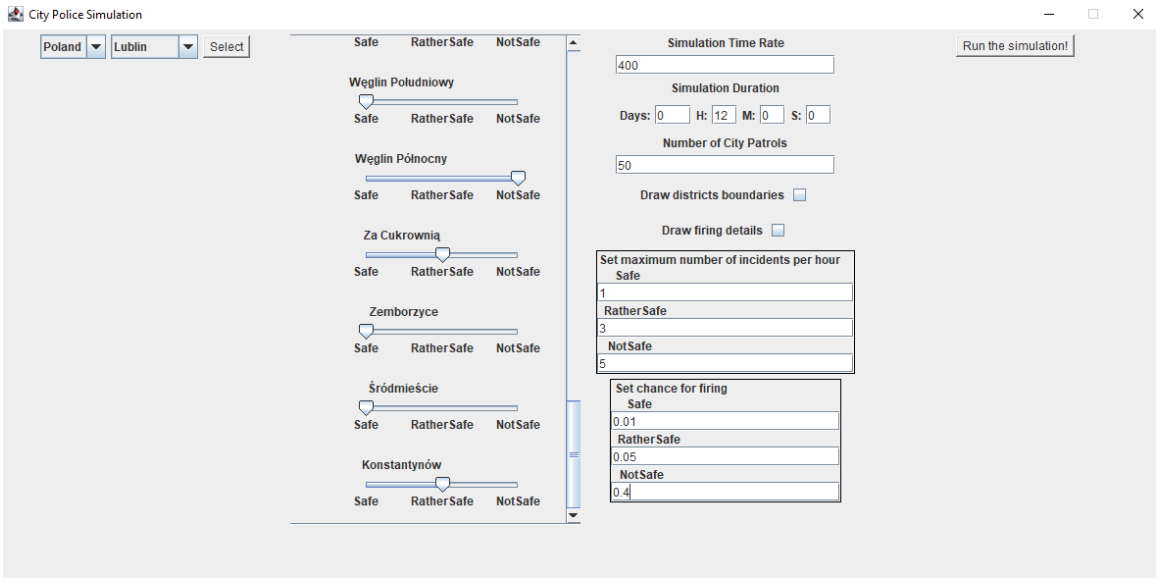


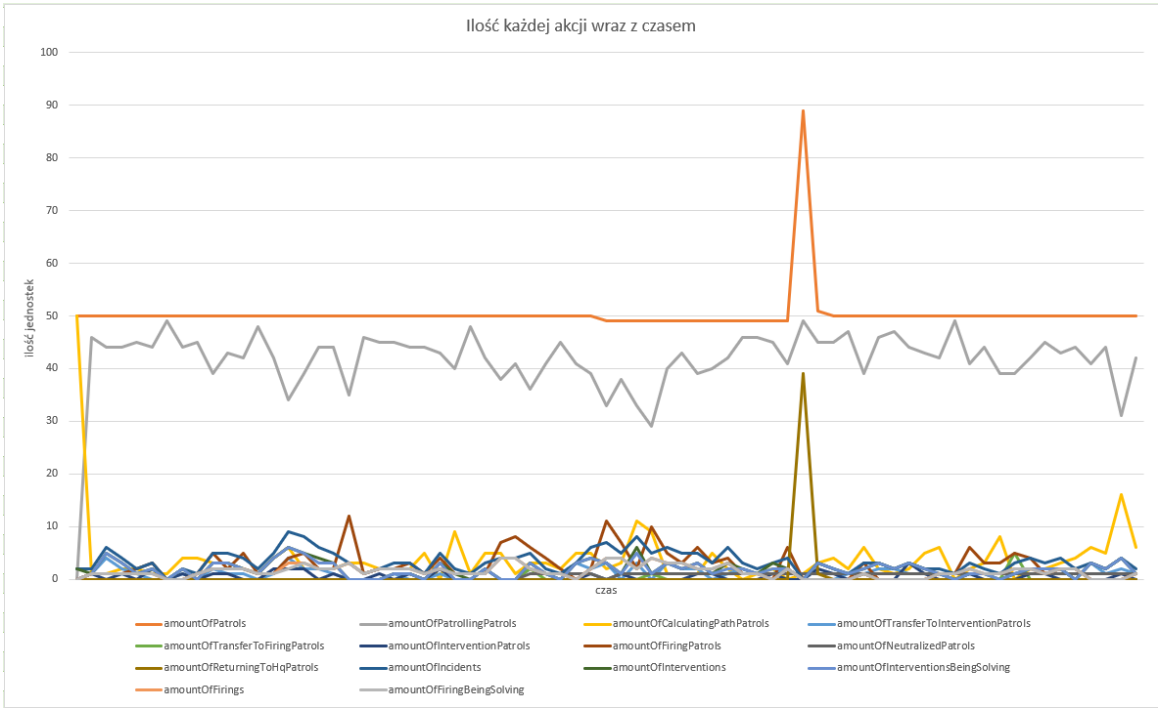
Kiedy symulacja się zakończy, a więc symulowany czas minie, pojawią się statystyki informujące o najważniejszych wydarzeniach w symulacji.



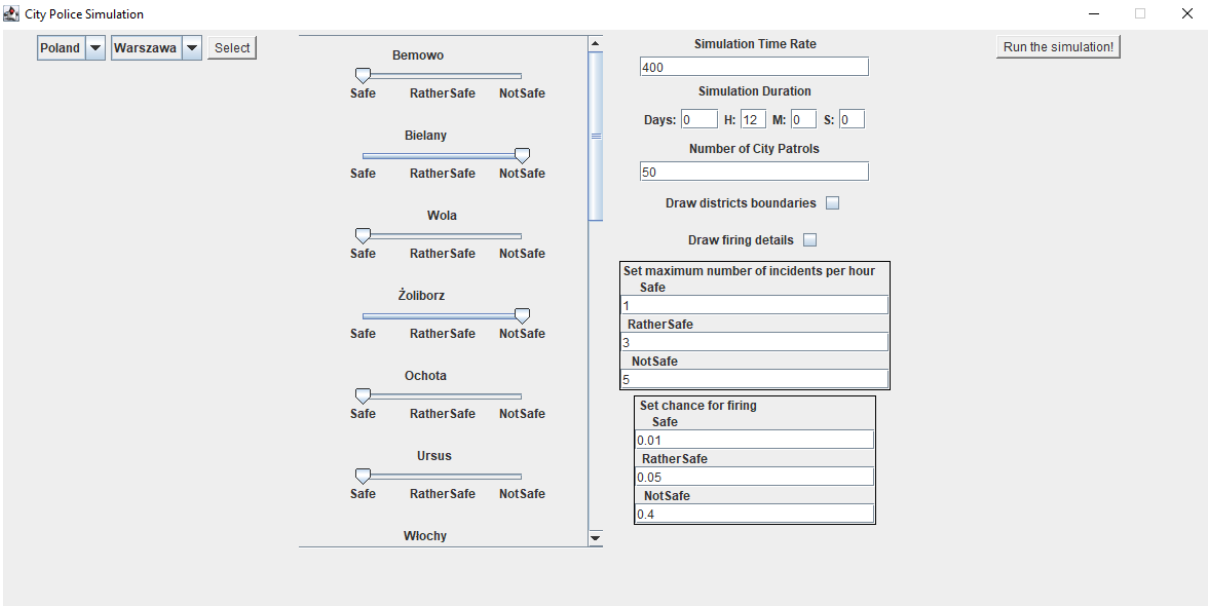
Przykładowe wyniki symulacji dla podanych konfiguracji:

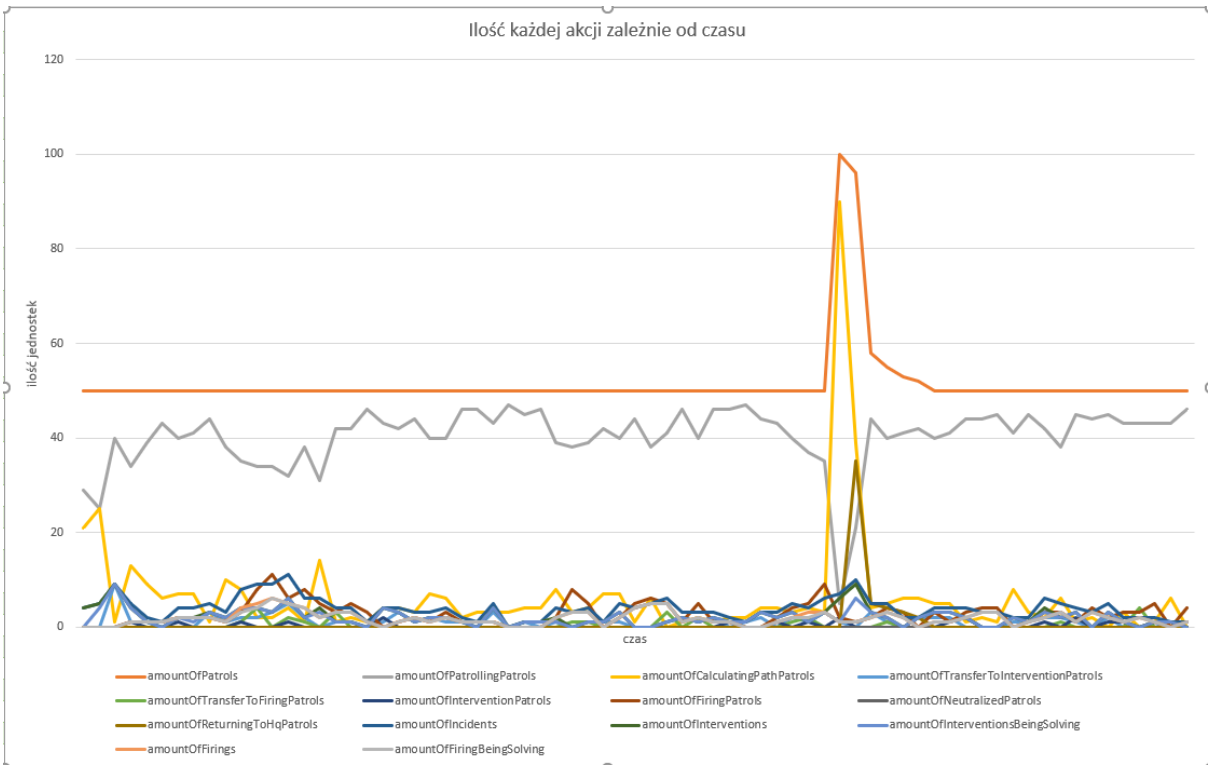
Lublin



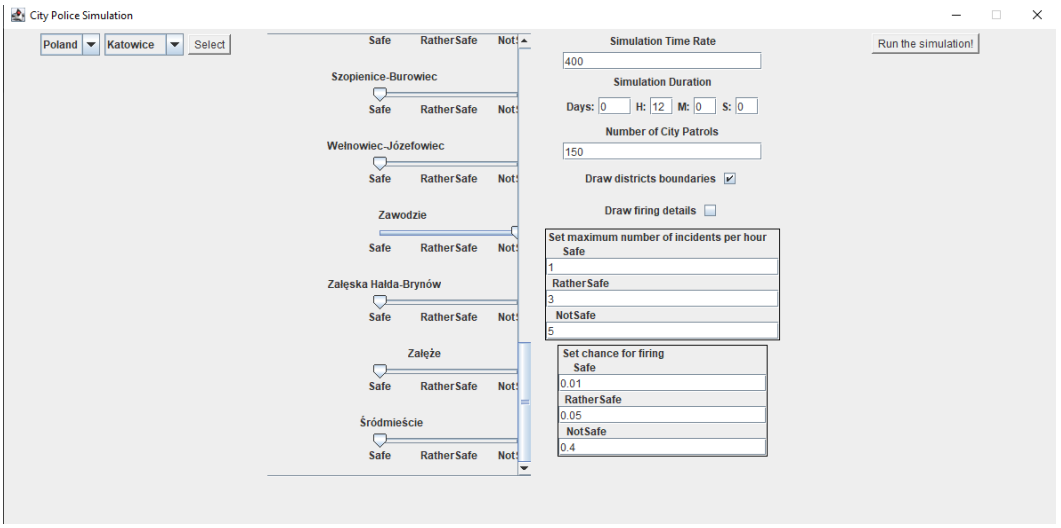

Ilość każdej akcji wraz z czasem
ilość jednostek
czas
amountOfPatrols
amountOfPatrollingPatrols
amountOfCalculatingPathPatrols
amountOfTransferToInterventionPatrols
amountOfTransferToFiringPatrols
amountOfInterventionPatrols
amountOfFiringPatrols
amountOfNeutralizedPatrols
amountOfReturningToHqPatrols
amountOfIncidents
amountOfInterventions
amountOfInterventionsBeingSolving
amountOfFirings
amountOfFiringBeingSolving

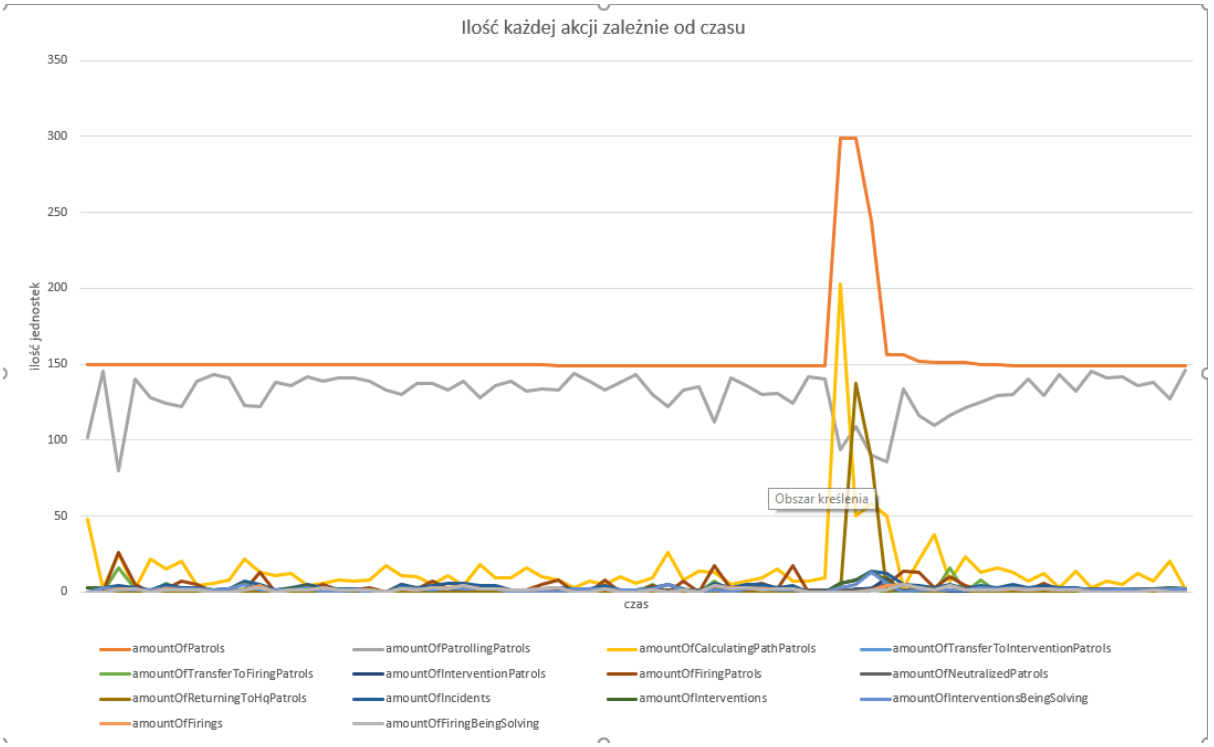
Warszawa


City Police Simulation
Poland
Warszawa
Select
Bemowo
Safe
RatherSafe
NotSafe
Bielany
Wola
Żoliborz
Ochota
Ursus
Włochy
Simulation Time Rate
400
Simulation Duration
Days: 0 H: 12 M: 0 S: 0
Number of City Patrols
50
Draw districts boundaries
Draw firing details
Set maximum number of incidents per hour
Safe
1
RatherSafe
3
NotSafe
5
Set chance for firing
Safe
0.01
RatherSafe
0.05
NotSafe
0.4
Run the simulation!


Ilość każdej akcji zależnie od czasu
120
100
80
60
40
20
0
ilość jednostek
czas
amountOfPatrols
amountOfPatrollingPatrols
amountOfCalculatingPathPatrols
amountOfTransferToInterventionPatrols
amountOfTransferToFiringPatrols
amountOfInterventionPatrols
amountOfFiringPatrols
amountOfNeutralizedPatrols
amountOfReturningToHqPatrols
amountOfIncidents
amountOfInterventions
amountOfInterventionsBeingSolving
amountOfFirings
amountOfFiringBeingSolving

Katowice


City Police Simulation
Poland
Katowice
Select
Safe
RatherSafe
Szopienice-Burowiec
Wełnowiec-Józefowiec
Zawodzie
Załęska Hałda-Brynów
Załęże
Śródmieście
Simulation Time Rate
400
Simulation Duration
Days: 0 H: 12 M: 0 S: 0
Number of City Patrols
150
Draw districts boundaries
Draw firing details
Set maximum number of incidents per hour
Safe
1
RatherSafe
3
NotSafe
5
Set chance for firing
Safe
0.01
RatherSafe
0.05
NotSafe
0.4
Run the simulation!



10. Zaplanowane i niezrealizowane funkcjonalności

Pierwszą niezrealizowaną funkcjonalnością jest tryb interaktywny, w którym można dodawać oraz usuwać interwencje. Zaimplementowanie go wymaga dużego nakładu czasu, którego zabrakło na wykończenie. W aplikacji jest możliwość wyboru kraju i miasta - na ten moment jest do dyspozycji tylko Polska oraz większość miast wojewódzkich Polski. Rozbudowa aplikacji o tą funkcjonalność jest mało czasochłonne, wymaga wprowadzenia zapytań do API o nowe miasta i ich dzielnice. W innym kraju administracja granic miasta może się różnić od Polski, co może być jednym utrudnieniem. Również nie dodano funkcjonalności ustawienia ilości jednostek policji dla każdej zmiany (aktualnie każda zmiana posiada tyle samo jednostek). Jeśli chodzi o założenia w symulacji, nie zostało wykonane zwiększenie lub zmniejszenie prawdopodobieństwa wystąpienia strzelaniny lub interwencji zależnie od pory dnia. Dodanie tej funkcjonalności wymaga przeredagowania części kodu, aby móc dostosować podział na dzień i noc i zmianę współczynników w odpowiednim czasie. Również prędkość poruszania się policjantów nie jest dostosowywana zależnie od aktualnego stanu na drodze (np. Prędkość poruszania się policji w stanie “patrolling” w czasie godzin szczytu) - dotyczy to zarówno jazdy na interwencję jak i jazdy na strzelaninę. Przed rozpoczęciem symulacji miała być możliwość dodania ilości jednostek policji na każdą zmianę - aktualnie jest możliwość wyboru liczby jednostek, jednak ta wartość jest taka sama dla wszystkich zmian.