Kraków 27.03.2023

Analiza problemu i dziedziny

Mikołaj Zasada, Patryk Studziński

1. Opis problemu

Ogólnym celem projektu jest poszerzenie istniejącego już symulatora HiPUTS (High Performance Urban Traffic Simulator) o nowe funkcjonalności wizualizacji. Obecne metody ze względu na swoje podejście nie pozwalają na wnikliwą analizę np. z nałożeniem pojazdów na mapę, a tylko na prostej grafowej reprezentacji grafowej która może być trudno interpretowalna. Nowa metoda ma umożliwiać dokładniejszą wizualną analizę wybranych fragmentów symulacji, ze szczególnym przyłożeniem wagi do szczegółowego badania działania symulacji np. na wielopoziomowych, wielopasmowych skrzyżowaniach autostrad, czy analogicznie skomplikowanym przykładom w bardziej zurbanizowanych fragmentach miasta. Ma to służyć w celu wyszukiwania błędów, sprawdzania poprawności, czy zwykłej obserwacji. Projekt ze względu na założenia dotyczące skali symulacji oraz idących za tym dużych ilości danych ją opisujących będzie wymagał odpowiedniego doboru narzędzi programistycznych oraz technologii pozwalających w krótkim czasie przetworzyć potencjalnie duże ilości danych.

2. Cele projektu

- Stworzenie aplikacji webowej umożliwiającej wizualizację wybranej jednej instancji obliczeniowej (jednego fragmentu mapy równoważne z przetwarzaniem danych od jednego uruchomionego procesu)
- Rozszerzenie obecnego symulatora o odpowiednie endpointy, które będą pozwalały na wykonywania celi wypisanych w następnych podpunktach
- Wstępnie wizualizacja ma działać w formie nanoszenia informacji z symulatora na mapę (sugerowana technologia leaflet), takich jak samochody, pasy, światła oraz inne, które w trakcie implementacji projektu okażą się możliwe i przydatne do zaimplementowania
- Możliwość zaznaczenia pojazdu na mapie w trakcie trwania wizualizacji (zapewne preferowanym sposobem będzie zaznaczenie kursorem myszki) w celu pozyskania dodatkowych informacji na jego temat, a także możliwości śledzenia jego dalszej podróży/symulacji.
- Możliwość przy zaznaczeniu sprawdzenia historii skąd i jaką drogą samochód przyjechał (konieczne sprawdzenie możliwości wykonania)
- Odpowiednie przetestowanie oprogramowania
- Zbadanie możliwości rozszerzenia na więcej niż jeden fragment mapy w symultanicznej wspólnej symulacji

- Zbadanie możliwości rozszerzenia symulacji o parametry płynności odtwarzania
- W zależności od powodzenia poprzednich podpunktów implementacja oraz dalszy rozwój projektu

3. Założenia projektu

Aplikacja ma przybrać formę strony w przeglądarce internetowej która w czasie rzeczywistym będzie odbierała informacje wysyłane z symulacji, odpowiednio przetwarzała i naniosła dane na mapę. Mapę tę użytkownik będzie mógł w dowolnym stopniu skalować oraz się po niej przemieszczać w celu obserwowania przemieszczających się pojazdów, zmieniających się świateł sygnalizacji i innych wspieranych obiektów. Ważna jest odpowiednia szczegółowość wizualizacji jak rozróżnianie pasów, krzywizny zakrętów i różnorodności skrzyżowań w tym wielokondygnacyjnych. Odpowiednie UI będzie wyświetlało w formie tabeli dodatkowe dane na temat zaznaczonych myszką obiektów. Prędkości wizualizacji przynajmniej wstępnie ograniczona jest przez prędkość symulacji domyślnie wynoszącą 1 step/s.

4. Potencjalne wyzwania

- Płynność animacji - Po wstępnej rozmowie z mgr. Mateuszem Najdkiem jako podstawowy cel został nam zasugerowany poziom odświeżania wysyłania informacji raz na sekundę. Takie podejście najprawdopodobniej eliminuje pierwotne obawy związane z przepustowością REST API. Natomiast w przypadku rozwijania możliwości wizualizacji także na wyższe odświeżania symulacji np. 30 razy na sekundę możliwe że będzie potrzebne rozważenie zmiany narzędzia w celu zwiększenia przepustowości. Jako że parametr jest ustawiany z poziomu konfiguracji startu symulacji trzeba zdecydować o przypadkach gdy symulacja będzie działać szybciej niż wizualizacja, a być może także w drugą stronę.
- Śledzenie pojazdu - Czysta wizualizacja symulacji nie wymaga żadnych dodatkowych informacji na temat obiektów które wyświetlamy poza typem, który może być dostarczony przez np. id listy, bądź podział API na różne endpointy od różnych obiektów. Zatem najlżejszym podejściem pod względem wysyłania danych jest wysyłanie samej pozycji na mapie. Oczywiście takie podejście praktycznie uniemożliwia śledzenie pojazdów gdyż między odświeżeniem nie przechowujemy żadnej informacji o tym kto jest kim, a tym bardziej gdzie zaraz będzie. Zatem wymagane będą pewne oznaczenia na etapie tworzenia wiadomości przez endpoint. Być może aby ograniczyć nakład danych, symulacja wysyłała by szczegółowe informacje na temat pojedynczego pojazdu z osobnego endpointa. A w rozwiniętej wersji na endpoincie były by ciągle dostępne

szczegółowa historia pojazdów.Dokładność reprezentacji na mapie - Problemem jest także dokładność reprezentacji na mapie pojazdów

- Historia zaznaczonego pojazdu - Aby być w stanie zobaczyć jaką drogę przebył pojazd od początku symulacji trzeba byłoby posiadać bardzo duży buffor co jest niewykonalne biorąc pod uwagę chęć optymalizacji szybkości działania i zmniejszenie nakładu danych na jeden krok wizualizacji. Jednym z możliwych podejść jest przechowywanie dokładnej infromacji 5-10 ostatnio zaznaczonych pojazdów co pozwalałoby na zaznaczenie przez użytkownika pojazdu albo określonej liczby pojazdów, oraz śledzenia ich na bieżąco aby uniknąć np. problemu z missclick podczas śledzenia danego pojazdu, zakładając że buffor historii były tworzony dopiero od jego zaznaczenia.
- Wizualizacja wielu procesów symulacji - Kolejnym problemem jest przechwytywanie informacji od wielu procesów zakładając że przechwytywanie informacji z jednego nie będzie problematyczne. Możliwe jest też utworzenie wersji wieloprocesorowej okrojonej w dodatkowe zalety w celu ulepszenia działania wielo procesowego.
- Dokładność wyświetlania elementów

5. Kroki tworzenia wizualizacji

Kryteria finalizacji można podzielić na parę podpunktów gdzie osiągnięcie każdego następnego jest bezpośrednio powiązane z jego problematyką.

1. Utworzenie wizualizacji przeglądarkowej z możliwością przesuwania i skalowania mapy
2. Rozszerzenie wizualizacji o różne obiekty, tak aby było możliwe ich sprawne rozróżnianie na mapie
3. Dodanie możliwości prostego śledzenia zaznaczonego pojazdu
4. Dodanie zapisywania historii śledzonego pojazdu od momentu jego zaznaczenia oraz wyświetlanie na jego temat bardziej szczegółowych informacji
5. Wyświetlanie szczegółowych informacji na temat wszelkich obiektów wizualizacji
6. Dodanie bufora w celu nie tracenia utworzonej już historii po odznaczeniu pojazdu (dodatkowo powinna ona być ciągle zbierana w tle)

Wybór narzędzi

Mikołaj Zasada, Patryk Studziński

Do wyświetlenia symulacji zdecydowaliśmy się wybrać język JavaScript oraz bibliotekę React. Ma ona wiele zalet w kontekście wyświetlenia symulacji.

Przede wszystkim posiada możliwość użycia gotowego komponentu React Leaflet do wyświetlenia mapy, która będzie podstawowym elementem w naszej symulacji.

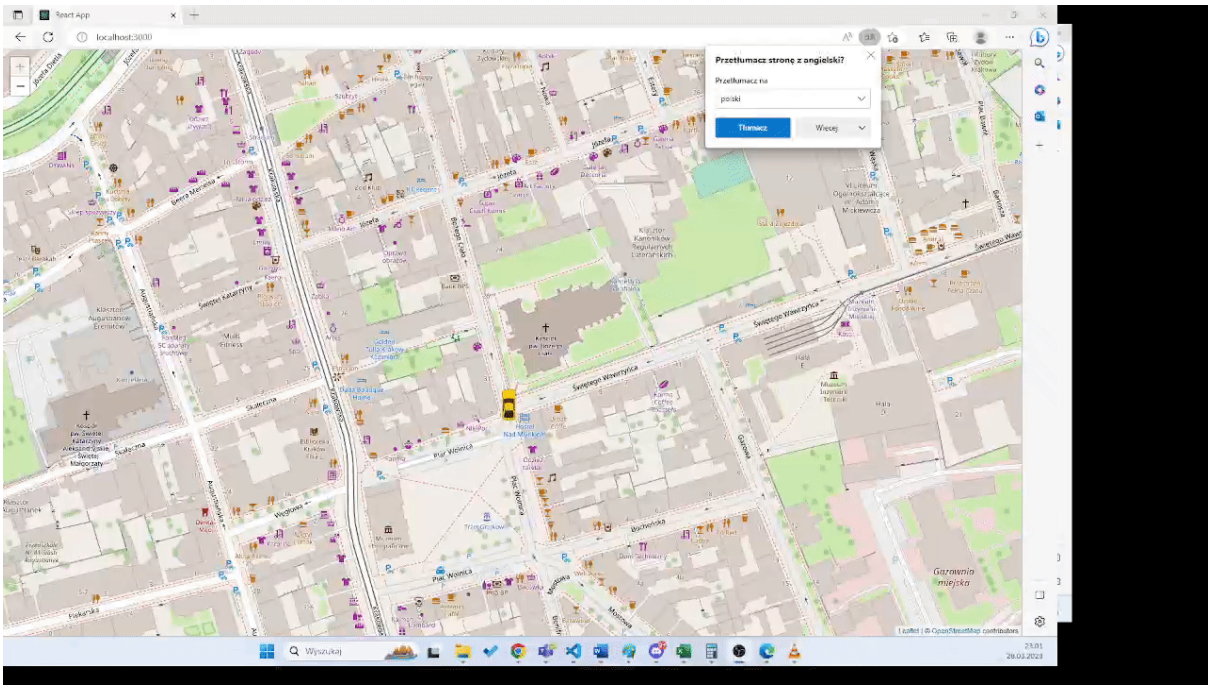
React opiera się na komponentach, dzięki czemu można stworzyć symulację z mniejszych i łatwiejszych elementów.

Cechuje się reaktywnością, pozwala na dynamiczne aktualizowanie elementów na podstawie zmiany stanu komponentów.

Jest również szybka, dzięki aktualizacji tylko tych komponentów których stan się zmienia, nie ma potrzeby przeładowywania za każdym razem wszystkich elementów.

Z racji popularności i dobrej dokumentacji łatwo można znaleźć rozwiązania napotkanych problemów.

Dla testów udało nam się stworzyć animację samochodu poruszającego się po drodze.



Język Java wybraliśmy do części serwerowej wizualizacji. Symulacja zrealizowana jest w Javie, z tego powodu w łatwy sposób będzie istniała możliwość dołożenia dodatkowego modułu do projektu i jego integracji z wizualizacją.