

Graphical editor for road network for autonomous traffic simulation

Edytor sieci drogowej na potrzeby symulacji ruchu autonomicznego

PAWEŁ LISIECKI *

DOI: <https://doi.org/10.17814/mechanik.2024.11.22>

This paper presents the issue of designing road networks for autonomous traffic simulation. Existing autonomous traffic engines are discussed, and their capabilities are compared with the requirements for high-immersion vehicle simulators. The extension to the Hermes editor for generating complex road network geometries is presented, along with tools to improve the work of road network designers. Finally, the advantages and disadvantages are outlined, and development possibilities for the designed solution are indicated.

KEYWORDS: real vehicle simulators, SUMO, OpenStreetMap, road networks, autonomous traffic simulation, traffic signal control

Artykuł przedstawia problematykę tworzenia sieci drogowych na potrzeby symulacji ruchu autonomicznego. Opisano istniejące silniki ruchu autonomicznego i zestawiono ich możliwości z wymaganiami stawianymi symulatorom rzeczywistych pojazdów o wysokim poziomie immersyjności. Przedstawiono rozszerzenie do edytora Hermes pozwalające na generowanie sieci drogowej o złożonej geometrii, z narzędziami usprawniającymi pracę projektanta sieci drogowej. Finalnie przedstawiono wady i zalety, a także wskazano możliwości rozwoju zaprojektowanego rozwiązania. **SŁOWA KLUCZOWE:** symulatory rzeczywistych pojazdów, SUMO, OpenStreetMap, sieci drogowe, symulacja ruchu autonomicznego, sterowanie sygnalizacją świetlną

Wprowadzenie

Symulatory rzeczywistych pojazdów odgrywają fundamentalną rolę w nowoczesnych metodach szkoleniowych, ponieważ oferują rozwiązania bezpieczne, przyjazne dla środowiska i korzystne ekonomicznie. Takie symulatory znajdują zastosowanie w ośrodkach badawczych oraz firmach i instytucjach, które szkolą początkujących bądź profesjonalnych kierowców.

Symulatory mogą mieć różne formy – od prostych symulatorów biurkowych po repliki całych kabin pojazdów z pełnym odwzorowaniem urządzeń pokładowych oraz systemem ruchu o sześciu stopniach swobody (6DOF), aby zapewnić maksymalny poziom immersji. Podstawową przewagą symulatorów nad tradycyjnymi metodami szkoleniowymi jest możliwość zasymulowania sytuacji awaryjnych, nietypowych czy niebezpiecznych, których odwzorowanie podczas tradycyjnego szkolenia byłoby bardzo kosztowne i/lub niebezpieczne. Szkolenie kierowców w niestandardowych warunkach w środowisku symu-

latora ma na celu zwiększenie bezpieczeństwa oraz podnoszenie umiejętności [1]. Symulatory pozwalają również na przeprowadzenie badań, które nie byłyby możliwe do zrealizowania w warunkach rzeczywistego ruchu drogowego [2–4].

W niektórych przypadkach wymaga się od symulatora wysokiego poziomu immersji oraz odwzorowania wirtualnego świata w celu symulowania sytuacji, które mogłyby się zdarzyć w prawdziwym świecie. Takie symulatory mają szczególną wartość dla ośrodków szkolących przyszłych maszynistów czy kierowców autobusów. Od prowadzącego pojazd kolejowy wymaga się dobrej znajomości szlaku, zanim będzie mógł samodzielnie poprowadzić pociąg. Z kolei od kierowcy autobusu oczekuje się dokładnej znajomości trasy, którą będzie obsługiwał.

Zapewnienie wysokiego poziomu immersji w symulatorach rzeczywistych pojazdów jest poważnym wyzwaniem, zwłaszcza gdy odwzorowuje się duży fragment świata, np. całe miasto czy całą trasę kolejową o długości kilkuset kilometrów. Wytworzenie dużych obszarów wirtualnego świata 3D o wysokim poziomie odwzorowania i szczegółowości jest złożone i czasochłonne. Problemy związane z modelowaniem świata 3D na potrzeby symulacji przedstawiono w [5].

Z podobnymi problemami trzeba się liczyć podczas modelowania sieci drogowej na potrzeby symulacji ruchu autonomicznego. Symulacja ruchu autonomicznego jest niezbędna do zapewnienia akceptowalnego poziomu immersji w symulatorach pojazdów osobowych, ciężarowych czy autobusów. W przypadku takich symulatorów wymagane jest dokładne odwzorowanie sieci drogowej całego miasta lub jego fragmentu (np. w pobliżu trasy autobusu). Problematykę projektowania sieci drogowej przedstawiono w [6] i [7].

Problem badawczy, który jest przedmiotem niniejszego, artykułu to opracowanie w ramach projektu narzędzia do projektowania sieci drogowej na potrzeby symulacji ruchu autonomicznego. Prezentowane narzędzia są częścią oprogramowania Hermes [5, 8], którego celem jest generowanie wirtualnego świata 3D na potrzeby symulacji. Dalsza część artykułu omawia różne dostępne narzędzia do symulacji ruchu autonomicznego, koncentrując się na analizie ich funkcjonalności oraz porównaniu z wymaganiami stawianymi symulatorom rzeczywistych pojazdów, w celu oceny istniejących rozwiązań pod kątem zastosowań w symulacji ruchu autonomicznego, oraz możliwościach integracji tych rozwiązań w edytorze Hermes.

* Mgr inż. Paweł Lisiecki – plisiecki@autocomp.com.pl, <https://orcid.org/0009-0002-0713-4434> – Autocomp Management Sp. z o.o., Szczecin, Polska

Silnik ruchu autonomicznego

Wiele jest narzędzi pozwalających na symulację ruchu autonomicznego, opartych na technologii agentowej [9]. Zastosowanie technologii agentowej jest niezwykle istotne w kontekście projektowania symulatorów jazdy, ponieważ pozwala na modelowanie zachowania indywidualnych uczestników ruchu drogowego. Ze wszystkich przedstawionych w pracy [9] narzędzi najbardziej uniwersalnym rozwiązaniem w kontekście symulowania ruchu autonomicznego jest SUMO (Simulation of Urban MObility). MATSim (Multi-Agent Transport Simulation) skupia się przede wszystkim na makroskopowych symulacjach, operując na uproszczonym modelu sieci drogowej, co jest niedopuszczalne z punktu widzenia symulatora rzeczywistych pojazdów. Polaris ma zaawansowane możliwości analizy i optymalizacji przepływu ruchu drogowego, ale nie oferuje pełnego wsparcia dla zaawansowanej symulacji mikroskopowej oraz szczegółowego odwzorowania dynamiki pojazdów w ruchu drogowym. SimMobility (Simulating Mobility), mimo swojej elastyczności w zakresie symulacji długoterminowej, średnioterminowej i krótkoterminowej, nie oferuje natywnej symulacji w czasie rzeczywistym. Biorąc pod uwagę możliwości i ograniczenia poszczególnych silników symulacji ruchu ulicznego, w dalszej części artykułu wszelkie problemy oraz proponowane rozwiązania będą zestawione z możliwościami i ograniczeniami symulatora ruchu autonomicznego SUMO.

Podczas tworzenia systemu edytora Hermes postawiono wymagania w stosunku do funkcjonowania ruchu autonomicznego na potrzeby realizacji przyszłych symulatorów. Wymagania te zestawiono z funkcjonalnościami SUMO. W wyniku analizy wyodrębniono wymagania, których natywnie nie wspiera SUMO, bądź wspiera je w ograniczonym zakresie. Należą do nich:

- dokładne odwzorowanie geometrii sieci drogowej za pomocą krzywych (niedopuszczalne jest użycie linii wielosegmentowej),
- symulacja ruchu kolumny pojazdów (na potrzeby symulatorów pojazdów uprzywilejowanych),
- sterowanie ruchem na skrzyżowaniu przez policjanta,
- symulacja i sterowanie ruchem autonomicznym w czasie rzeczywistym,
- symulacja korytarza życia w reakcji na pojawienie się pojazdu uprzywilejowanego.

W SUMO sieć drogowa składa się z linii wielosegmentowej, co w niektórych przypadkach jest niewystarczające do realistycznego odwzorowania sieci drogowej. Dodatkowo SUMO nie wspiera natywnie funkcjonalności sterowania ruchem pojazdów na skrzyżowaniu przez policjanta. Symulacje korytarza życia oraz kolumny pojazdów uprzywilejowanych są możliwe do zrealizowania za pośrednictwem interfejsu komunikacyjnego TraCI (Traffic Control Interface), jednak wymagają pewnego dostosowania na potrzeby symulatora – np.: wstawianie kolumny pojazdów w symulowanym świecie, konfiguracja trasy przejazdu, konfiguracja tzw. pilotów kolumny pojazdów czy konfiguracja odpowiedniej przestrzeni w kolumnie

na pojazd symulowany (czyli pojazd, którym steruje użytkownik symulatora).

Ważnym aspektem symulacji ruchu autonomicznego jest działanie w czasie rzeczywistym przy minimalnych opóźnieniach (maksymalnie 30 ms). SUMO pozwala na konfigurację symulacji w taki sposób, aby przebiegała ona w czasie rzeczywistym, jednak rozszerzenie funkcjonalności SUMO wymaga integracji z interfejsem TraCI, co wprowadza dodatkową warstwę komunikacji między silnikiem symulacji a silnikiem ruchu autonomicznego. To z kolei implikuje wprowadzenie dodatkowego opóźnienia w torze komunikacyjnym między silnikiem symulacji a silnikiem ruchu autonomicznego. Optymalnym rozwiązaniem jest zastosowanie tylko jednej warstwy komunikacji, a właściwie bezpośredniej komunikacji między silnikiem ruchu autonomicznego a silnikiem symulacji. Wynikiem analizy możliwości SUMO w kontekście spełnienia wymagań była decyzja o stworzeniu autorskiego rozwiązania w zakresie symulacji ruchu autonomicznego na potrzeby symulatorów rzeczywistych pojazdów.

Problem tworzenia sieci drogowej na potrzeby symulacji ruchu autonomicznego

Narzędzie do symulacji ruchu autonomicznego generuje dane o ruchu autonomicznym na podstawie informacji z wirtualnej reprezentacji sieci drogowej. Sieć drogowa musi zostać zamodelowana z użyciem odpowiedniego edytora lub zostać wygenerowana automatycznie. Automatyczna generacja sieci drogowej jest dobrym rozwiązaniem z punktu widzenia czasochłonności, zwłaszcza gdy modelowana jest sieć drogowa na dużym obszarze lub dotyczy gęsto zaludnionego miasta (ze względu na liczbę ulic). Wadą takiego rozwiązania jest utrata dokładności modelowanej sieci drogowej z powodu wysokiej wrażliwości algorytmów automatyzujących generację na niestandardowe obszary sieci drogowej. Dodatkowo potrzebne są odpowiednie dane wejściowe do wygenerowania takiej sieci drogowej.

Narzędzie SUMO pozwala na wykorzystanie danych OSM (OpenStreetMap) do wygenerowania danych o ścieżkach ruchu, znakach i sygnalizacji świetlnej. Niestety dane OSM są tworzone przez szeroką społeczność, w związku z czym są podatne na błędy, co pokazano w [10]. Dodatkowo w danych OSM często brakuje informacji o rzeczywistej geometrii danego fragmentu drogi – dostępna jest tylko uproszczona reprezentacja, co również ogranicza zakres stosowności tego narzędzia do generowania ścieżek dla ruchu autonomicznego. Na rys. 1 pokazano, że oprócz prostych skrzyżowań sieć drogowa zawiera często złożone skrzyżowania, które ciężko byłoby wygenerować w sposób automatyczny.

Ze względu na wymienione ograniczenia automatycznego generowania sieci drogowej z użyciem SUMO oraz samych danych OSM, zdecydowano się na wprowadzenie własnego formatu opisu sieci drogowej. W związku z tym, że zaprojektowany format ma postać tekstową, możliwe jest ręczne tworzenie sieci



Fig. 1. Different types of intersections in the road network: a) a simple four-entry intersection, b) a complex canalized intersection with traffic lights and tram lines

Rys. 1. Różne rodzaje skrzyżowań w sieci drogowej: a) proste skrzyżowanie czterowlotowe, b) złożone skrzyżowanie skanalizowane z sygnalizacją świetlną i liniami tramwajowymi

drogowej poprzez dodawanie kolejnych wpisów. Taka metoda jest obciążona licznymi wadami, z których największą jest podatność na błędy i brak możliwości sprawdzenia poprawności sieci bez uruchomienia symulacji ruchu autonomicznego. W związku z tym zdecydowano się stworzyć własny edytor ścieżek dla ruchu autonomicznego. Narzędzie zostało zintegrowane w edytorze Hermes.

Edytor sieci drogowej

W pracy [8] przedstawiono funkcjonalności edytora w zakresie generacji terenu przestrzennego. Na potrzeby modelowania sieci drogowej dla symulacji ruchu autonomicznego wykorzystano narzędzie importu oraz prezentacji danych rastrowych dostępne w edytorze. Narzędzie to umożliwia rysowanie ścieżek dla ruchu autonomicznego na podkładzie rastrowym. Podkład tworzony jest na podstawie danych przestrzennych, które w pełni odzwierciedlają rzeczywistość. Dane rastrowe mogą być pobierane z dowolnych serwisów internetowych udostępniających usługi WMS, WMTS oraz WCS.

Opis sieci drogowej

Podstawowymi elementami modelowanej infrastruktury drogowej przeznaczonymi dla ruchu autonomicznego są drogi oraz skrzyżowania. Droga jest abstrakcją agregującą pasy ruchu i sama nie ma fizycznej reprezentacji w sieci. Podobnie skrzyżowanie jest abstrakcją, która łączy drogi, a właściwie pasy ruchu ze sobą za pośrednictwem korytarzy. Fizyczną reprezentację mają pasy ruchu oraz korytarze, które łączą dwa pasy ruchu na skrzyżowaniu.

Kolejnymi elementami sieci drogowej są znaki. Znaki w zależności od swojej kategorii mogą oddziaływać na funkcjonowanie ruchu autonomicznego, np. znaki ograniczenia prędkości czy znaki stopu. W większości przypadków znaki nie mają wpływu na działanie ruchu autonomicznego, ponieważ reguły definiowane są

na poziomie metadanych dróg, pasów ruchu, korytarzy i skrzyżowań.

Ostatnim elementem sieci drogowej jest sygnalizacja świetlna. Składa się ona z dwóch części – wizualnej oraz logicznej. Część wizualna definiuje wygląd sygnalizatora oraz jego umieszczenie w świecie wirtualnym. Część logiczna opisuje zasady działania sygnalizacji świetlnej na skrzyżowaniu. Sygnalizator świetlny można powiązać z co najmniej jednym korytarzem. Diagram zależności pomiędzy poszczególnymi elementami sieci drogowej przedstawiono na rys. 2.

W związku z wymaganiami odnośnie do pełnego odwzorowania geometrii sieci drogowej, wprowadzono w edytorze możliwość tworzenia różnych rodzajów krzywych. Pasy ruchu oraz korytarze mogą być reprezentowane jako złożone obiekty geometryczne

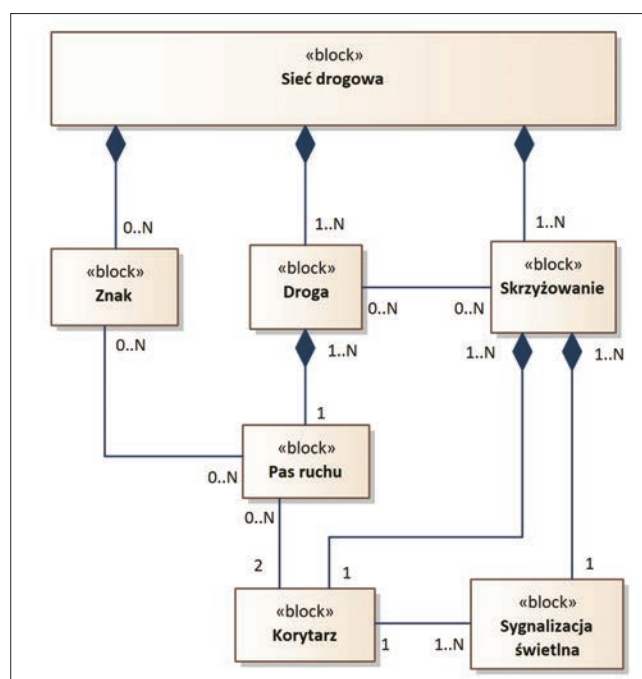


Fig. 2. Diagram of the relationships between the elements of the road network

Rys. 2. Diagram zależności między elementami sieci drogowej

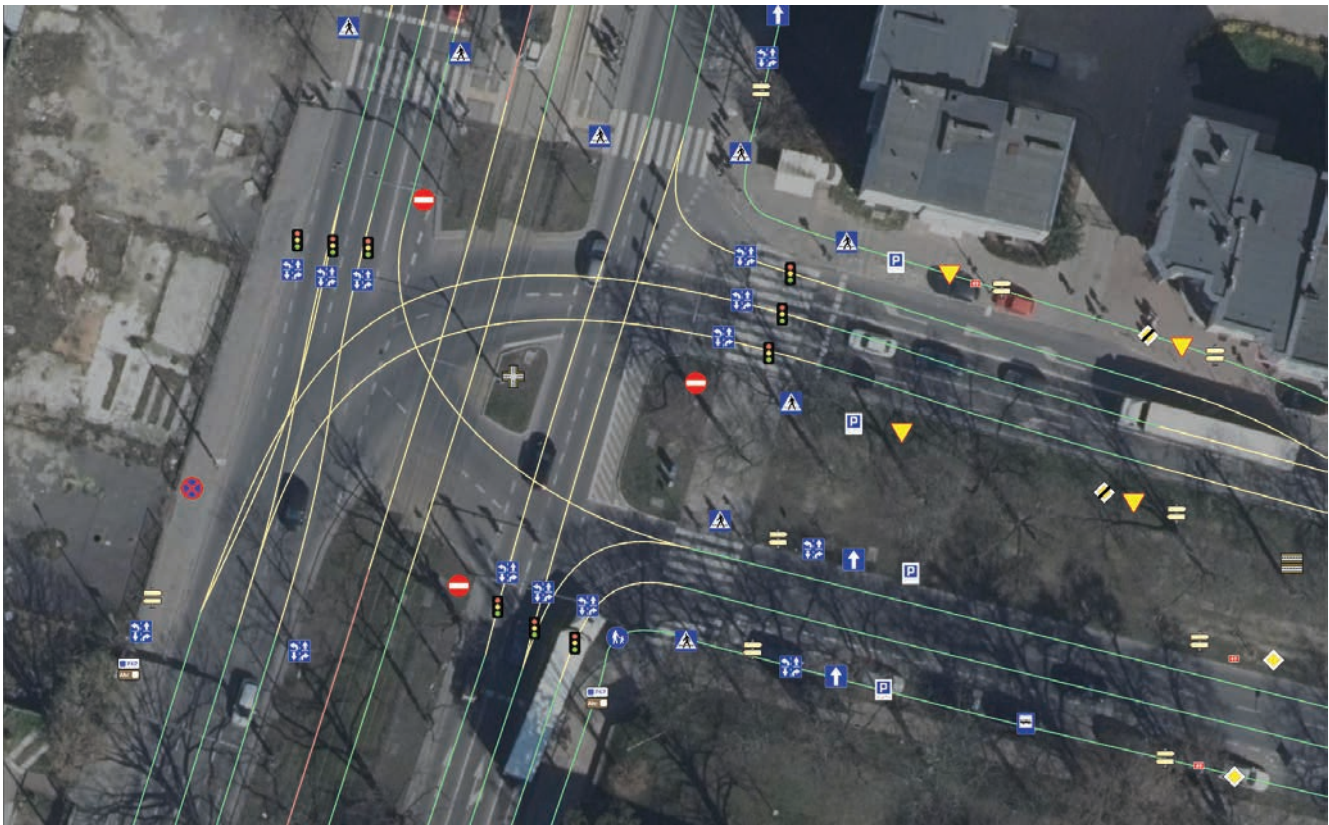


Fig. 3. Road network created in the Hermes editor
Rys. 3. Sieć drogowa stworzona w edytorze Hermes

składające się z wielu segmentów, z których każdy może przyjąć formę odcinka prostego, krzywej Bézie-
ra lub krzywej Catmull-Rom. Każdy z tych segmentów pozwala na dokładne odwzorowanie różnorodnych kształtów drogi, zarówno w przypadku prostych, jak i bardziej skomplikowanych zakrętów i połączeń. Efekt działania edytora przedstawiono na rys. 3.

Reguły ruchu drogowego

Podstawowymi regułami ruchu drogowego, które muszą zostać zamodelowane, są zasady pierwszeństwa przejazdu oraz reguły określające możliwość przejazdu przez daną kategorię pojazdów przez dany pas ruchu bądź korytarz. Pierwszeństwo przejazdu definiowane jest na poziomie skrzyżowań. Dla każdego korytarza należy zdefiniować zasady zachowania pojazdów względem pozostałych korytarzy należących do tego samego skrzyżowania. Możliwe jest zdefiniowanie jednego z trzech zachowań: „ustąp pierwszeństwa”, „pierwszeństwo nad korytarzem” lub „nieistotne”. Ostatnie zachowanie jest definiowane dla pary korytarzy, które nie przecinają się na skrzyżowaniu. Dodatkowo możliwe jest zdefiniowanie niezależnych zasad pierwszeństwa, kiedy na skrzyżowaniu działa sygnalizacja świetlna oraz kiedy nie działa (lub na danym skrzyżowaniu nie występuje). Konfigurator pierwszeństwa przedstawiono na rys. 4.

Na rzeczywistych drogach kierujący może się poruszać tylko po określonych pasach ruchu, które są odpowiednie dla typu pojazdu, którym kieruje. Na przykład kierowca samochodu osobowego nie może się poruszać po ścieżce rowerowej. Zdarza się również,

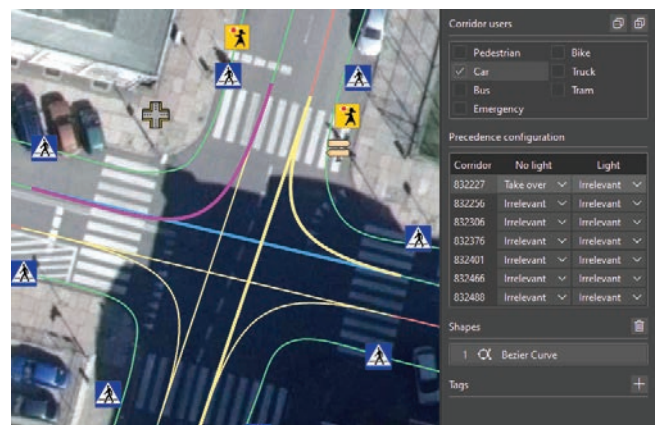


Fig. 4. Corridor configurator – the configured corridor is marked in pink, the corridor for which the priority rule is defined in blue

Rys. 4. Konfigurator korytarza – na różowo oznaczono korytarz konfigurowany, na niebiesko oznaczono korytarz, dla którego definiowana jest reguła pierwszeństwa

że jeden pas ruchu jest dostępny dla kilku typów pojazdów – np. na pasie ruchu znajdują się zarówno tory tramwajowe, jak i buspas. W edytorze sieci drogowej możliwe jest zdefiniowanie dowolnej kombinacji typów pojazdów dla każdego z pasów ruchu oraz korytarzy (prawy górny róg na rys. 4).

Narzędzie multibrush

Sieć drogowa składa się często z długich odcinków, które nie są przecinane żadnym skrzyżowaniem. Taka sytuacja występuje zazwyczaj na autostradach i drogach szybkiego ruchu, ale także na głównych trasach

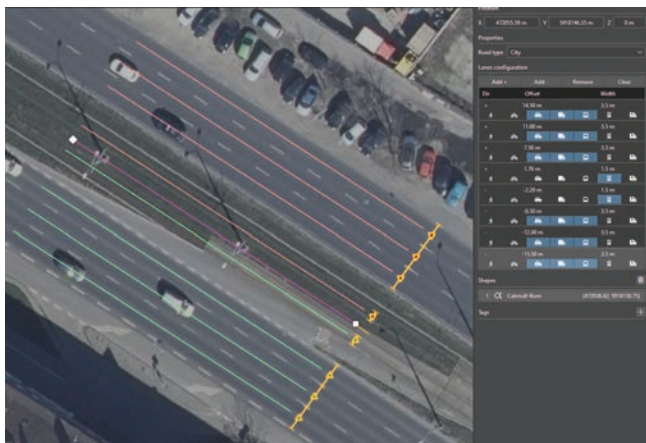


Fig. 5. The multibrush tool that draws multiple lanes at the same time (traffic lanes in a given direction are marked in colors)
Rys. 5. Narzędzie *multibrush* rysujące wiele pasów ruchu jednocześnie (kolorami zaznaczono pasy ruchu w danym kierunku)

tranzytowych w dużych miastach. W takim przypadku mamy do czynienia z drogą dwukierunkową wielopasmową, często przedzieloną pasem zieleni.

W przypadku klasycznej drogi ekspresowej o dwóch pasach ruchu konieczne jest wymodelowanie czterech krzywych o identycznej bądź bardzo zbliżonej geometrii. W celu przyspieszenia tworzenia takich fragmentów trasy zaprojektowano narzędzie o nazwie *multibrush* (wielopędzel), które pozwala na wprowadzenie wszystkich pasów ruchu w ramach wspólnej drogi jednocześnie. Dla danej drogi edytor pozwala na zdefiniowanie odpowiedniej liczby pasów ruchu w obu kierunkach, określenie ich szerokości, odległości od środka drogi oraz wskazanie typów pojazdów, które mogą się poruszać poszczególnymi pasami ruchu. Podczas wstawiania kolejnych punktów na scenę 2D wszystkie pasy rysują się automatycznie, zgodnie ze zdefiniowanymi regułami, zachowując odpowiednią geometrię. Narzędzie można wykorzystać do rysowa-

nia dowolnej drogi o co najmniej dwóch pasach ruchu, co znacznie przyspiesza proces tworzenia sieci drogowej. Efekt działania narzędzia *multibrush* przedstawiono na rys. 5.

Kreator znaków

W artykule [5] przedstawiono, że najbardziej czasochłonnym etapem pracy w procesie wytwarzania trójwymiarowego świata na potrzeby symulacji jest etap modelowania świata 3D. Podczas tworzenia edytora dla sieci drogowej postanowiono usprawnić ten proces w kontekście wstawiania w wirtualnym świecie znaków i sygnalizatorów świetlnych. Dane o znakach oraz sygnalizatorach, zawarte w pliku wynikowym opisującym sieć drogową, są wykorzystywane przez narzędzia do generowania świata 3D do automatycznego osadzania tych elementów w świecie symulatora. Takie rozwiązanie pozwala przyspieszyć generowanie wirtualnego świata 3D. Jednocześnie takie podejście umożliwia ponowne wykorzystanie tych samych danych (o znakach i sygnalizatorach świetlnych), by uniknąć duplikacji informacji i związanych z tym problemów.

W edytorze możliwe jest umieszczenie znaku na scenie oraz zdefiniowanie jego kategorii. Do niektórych typów znaków edytor oferuje konfigurator (rys. 6) pozwalający na zdefiniowanie dowolnego wskazania na znaku. W przypadku sygnalizacji możliwe jest określenie konkretnego typu sygnalizatora, np.: sygnalizator dla pieszych, sygnalizator ze światłem bezkolizyjnym czy sygnalizacja dla autobusu.

Konfigurator kontrolera sygnalizacji świetlnej

Kolejnym elementem edytora sieci drogowej jest konfigurator kontrolera sygnalizacji świetlnej na skrzyżowaniu. Sterowanie sygnalizacją świetlną odbywa się za pośrednictwem kontrolera, który zarządza cyklami świateł. Dla jednego skrzyżowania możliwe jest zdefiniowanie kilku różnych cykli, np. dla godzin porannych czy cyklu awaryjnego (migającego światła żółtego). W każdym cyklu możliwe jest zdefiniowanie poszczególnych faz świateł oraz czasu ich trwania. Konfigurator pozwala na zdefiniowanie kontrolera za pośrednictwem graficznego interfejsu użytkownika. Konfiguracja kontrolera przebiega w następujących krokach:

- 1) Wybór skrzyżowania, na którym ma zostać skonfigurowany kontroler.
- 2) Dodawanie i definiowanie faz świateł – określenie czasu trwania i koloru poszczególnych świateł w danej fazie.
- 3) Opcjonalnie – zdefiniowanie kolejnego programu świateł.

Ponieważ niektóre fazy świateł mogą się powtarzać, konfigurator pozwala na kopiowanie danej fazy lub zreferowanie już istniejącej jako nowej fazy – dzięki takiemu rozwiązaniu możliwe jest konfigurowanie kilku faz jednocześnie. Konfigurator umożliwia zdefiniowanie policjanta, który może sterować ruchem na skrzyżowaniu. Dla policjanta możliwa jest do ustawienia animacja, położenie oraz rotacja. W celu sprawdzenia poprawności działania kontrolera sygnalizacji



Fig. 6. Sign configurator (a) and sign representation in the virtual world (b)
Rys. 6. Konfigurator znaków (a) i reprezentacja znaku w świecie wirtualnym (b)

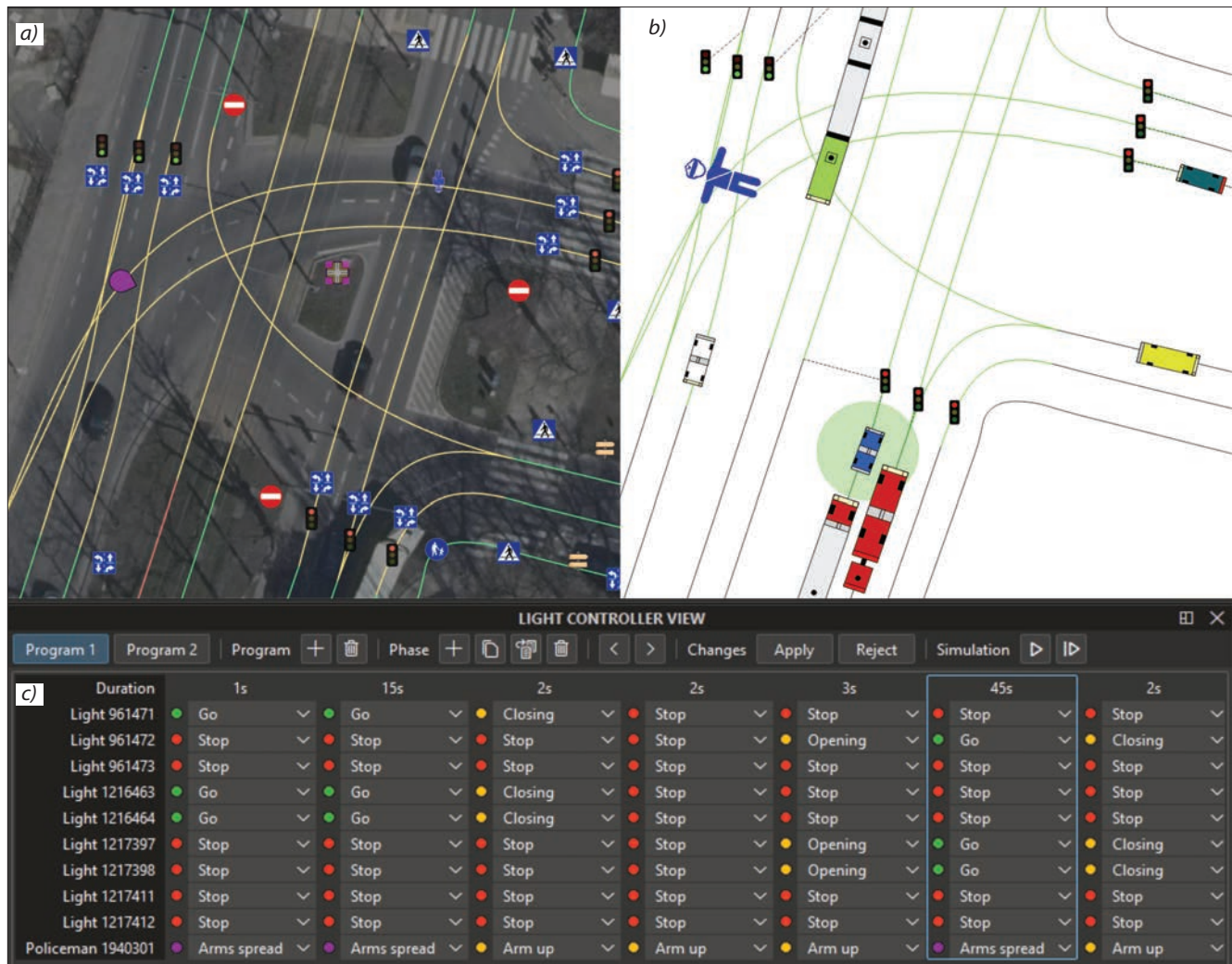


Fig. 7. Layout of the modeled intersection (a), presentation of the operation of the designed intersection (b) and light controller configurator (c)
Rys. 7. Układ modelowanego skrzyżowania (a), prezentacja działania zaprojektowanego skrzyżowania (b) oraz konfigurator kontrolera świateł (c)

światelnej konfigurator pozwala na zasymulowanie działania danego cyklu z rzeczywistym czasem trwania poszczególnych faz. Podczas symulacji na widoku sceny 2D sygnalizatory świetlne zmieniają swoją reprezentację graficzną zgodnie z aktualnie symulowaną fazą. Na rys. 7 pokazano widok konfiguratora kontrolera sygnalizacji świetlnej oraz efekt działania zamodelowanej sieci drogowej na podglądzie silnika ruchu autonomicznego.

Podsumowanie

W artykule przedstawiono problematykę tworzenia sieci drogowej na potrzeby symulacji ruchu autonomicznego w symulatorach rzeczywistych pojazdów o wysokim poziomie immersji. Omówiono wymagania, jakie stawia się silnikowi symulacji ruchu autonomicznego w symulatorze, i zestawiono je z możliwościami istniejących silników symulujących ruch pojazdów. Uzasadniono decyzję o stworzeniu własnego silnika ruchu autonomicznego oraz o rozszerzeniu funkcjonalności edytora Hermes o moduł tworzenia sieci drogowej. Przedstawiono funkcjonalności stworzonego modułu ze wskazaniem na spełnienie wymagań postawionych silnikowi ruchu autonomicznego. Przedstawiono narzędzia usprawniające pracę oraz

zmniejszające ryzyko powstania błędów w procesie projektowania sieci drogowej.

Stworzone narzędzie zostało zintegrowane w edytorze Hermes, co pozwala na wykorzystanie jednego programu do modelowania świata 3D oraz sieci drogowej. Dzięki takiemu rozwiązaniu możliwe jest ponowne użycie tych samych danych (np. o znakach drogowych) w procesie projektowania świata na potrzeby symulatora. Niezwykle przydatnymi narzędziami przyspieszającymi pracę projektanta układu dróg są narzędzie *multibrush* umożliwiające szybkie tworzenie wielopasmowych dróg oraz konfigurator kontrolera sygnalizacji świetlnej, który pozwala na elastyczną konfigurację sygnalizacji świetlnej na skrzyżowaniu. Kolejną zaletą edytora jest możliwość dokładnego odwzorowania geometrii drogi, co jest niewykonalne z wykorzystaniem edytora dostarczanego przez SUMO.

Mimo licznych zalet prezentowane rozwiązanie ma pewne ograniczenia. Ręczne modelowanie sieci drogowej może być czasochłonne, zwłaszcza w przypadku odwzorowywania dużych obszarów. W przyszłości możliwe jest rozszerzenie modułu o procedury automatycznego generowania sieci drogowej, co znacząco skróciłoby czas potrzebny na modelowanie. Takie rozwiązanie zaproponowano w [7]. Własny format opisu sieci drogowej, który nie jest kompatybilny z żadnym

istniejącym formatem, również stanowi wyzwanie, ograniczając stosowalność edytora do zaprojektowanego, autorskiego silnika ruchu autonomicznego. Rozszerzalny moduł eksportu danych w edytorze Hermes daje jednak możliwość rozwoju w przyszłości narzędzia eksportu, które wygeneruje dane w formacie zgodnym z SUMO, co pozwoliłoby na połączenie obu systemów i stworzyło nowe możliwości w zakresie badań nad symulacją ruchu autonomicznego.

Podziękowania

Zaprezentowane w artykule tezy oraz wyniki stanowią rezultat realizacji projektu: „Innowacyjna technologia tworzenia środowiska dla różnego typu symulatorów pojazdów lądowych i wodnych”, realizowanego przez Autocomp Management w ramach Regionalnego Programu Operacyjnego Województwa Zachodniopomorskiego 2014–2020, Oś Priorytetowa 1: Gospodarka, Innowacje, Nowoczesne Technologie, Działanie 1.1 Projekty badawczo-rozwojowe przedsiębiorstw, typ projektu 2: Projekty badawczo-rozwojowe przedsiębiorstw ukierunkowane na wdrożenie wyników prac B+R w działalności gospodarczej.

BIBLIOGRAFIA

- [1] Alonso F., Faus M., Riera J.V., Fernandez-Marin M., Useche S.A. “Effectiveness of driving simulators for drivers’ training: a systematic review”. *Applied Sciences*. 13, 9 (2023): 5266, <https://doi.org/10.3390/app13095266>.
- [2] Soares S., Lobo A., Ferreira S., Cunha L., Couto A. “Take-over performance evaluation using driving simulation: a systematic review and meta-analysis”. *European Transport Research Review*. 13, 47 (2021): 1–18, <https://doi.org/10.1186/s12544-021-00505-2>.
- [3] Yadav A.K., Khanuja R.K., Velaga N.R. “Gender differences in driving control of young alcohol-impaired drivers”. *Drug Alcohol Depend.* 213 (2020): 108075, <https://doi.org/10.1016/j.drugalcdep.2020.108075>.
- [4] Čulík K., Kalašová A., Štefancová V. “Evaluation of driver’s reaction time measured in driving simulator”. *Sensors*. 22, 9 (2022): 3542, <https://doi.org/10.3390/s22093542>.
- [5] Lisiecki P., Szłapczyński M., Chołodowicz E. „Automatyczna generacja wirtualnych elementów infrastruktury kolejowej”. *Pomiary Automatyka Robotyka*. 28, 2 (2024): 119–124, https://doi.org/10.14313/PAR_252/119.
- [6] Zhuoxiao Meng, Xiaorui Du, Sottovia P. et al. “Topology-Preserving Simplification of OpenStreetMap Network Data for Large-scale Simulation in SUMO”. *SUMO Conference Proceedings*. (2022): 181–197.
- [7] Muktadir G.M., Jawad A., Paranjape I., Whitehead J., Shepelev A. “Procedural generation of high-definition road networks for autonomous vehicle testing and traffic simulations”. *SAE International Journal of Connected and Automated Vehicles*, 6, 12-06-01-0007 (2022): 99–120, <https://doi.org/10.4271/12-06-01-0007>.
- [8] Teclaw M., Szłapczyński M. „Generacja terenu przestrzennego za pomocą edytora Hermes”. *Pomiary Automatyka Robotyka*. 28, 2 (2024): 113–118, https://doi.org/10.14313/PAR_252/113.
- [9] Nguyen J., Powers S.T., Urquhart N., Farrenkopf T., Guckert M. “An overview of agent-based traffic simulators”. *Transp Res Interdiscip Perspect.* 12 (2021): 100486, <https://doi.org/10.1016/j.trip.2021.100486>.
- [10] Tabet F., Pentyala S., Patel B.H. et al. “Osmrunner: A system for exploring and fixing OSM connectivity”. *22nd IEEE International Conference on Mobile Data Management (MDM)*. Toronto, Canada. (2021): 193–200, <https://doi.org/10.1109/MDM52706.2021.00039>. ■