

PLAKAT INFORMACYJNY PROJEKTU GRUPOWEGO – MAJ 2024

KATEDRA ALGORYTMÓW I MODELOWANIA SYSTEMÓW

Zespół projektowy: 4@KAMS'2023/24	1. Łukasz Nowakowski - kierownik 2. Konrad Bryłowski 3. Aleksander Czerwionka 4. Michał Krause 5. Krystian Nowakowski
Opiekun:	dr inż. Paweł Kowalski
Klient:	dr inż. Krzysztof Manuszewski
Data zakończenia:	26.05.2024
Słowa kluczowe:	pojazd autonomiczny, symulator, SI, unity



TEMAT PROJEKTU:

Symulator pojazdu autonomicznego

CELE I ZAKRES PROJEKTU:

Celem projektu jest wytworzenie symulatora, który będzie przydatny i łatwy w użyciu przy trenowaniu sztucznej inteligencji do sterowania małymi pojazdami bez potrzeby uruchamiania jej na modelu rzeczywistym. Przewidziane są dwa tryby nauki – naśladowanie operatora oraz samodzielną naukę – oraz dwa środowiska treningu – platformę ze ścieżką do śledzenia oraz scenerię korytarza WETI.

OSIĄGNIĘTE REZULTATY:

Po dwóch semestrach realizacji projektu wytworzono symulator w Unity zawierający generator losowej platformy ze ścieżką oraz scenerią pierwszego piętra budynku NE. W symulatorze zamodelowano pojazd wydrukowany w drukarce 3D przez opiekuna projektu, model przy domyślnych parametrach symulacji zachowuje się w sposób zbliżony do rzeczywistego, zaimplementowano zbieranie danych o sile napędu oraz kącie skrętu kół, a także obrazów otoczenia modelu. Zaimplementowano także interfejs komunikacji w postaci serwera WWW z API pozwalającym na inicjowanie nowych map oraz pojazdów na mapie, a także sterowanie parametrami pojazdu: mocą silnika oraz kątem skrętu. Wszystkie te funkcje dostępne są także przez interfejs użytkownika. Zgodnie z zaplanowaną organizacją pracy, postępy projektu są publicznie dostępne na platformie GitHub: <https://github.com/autonomous-vehicle-sim/Simulator>

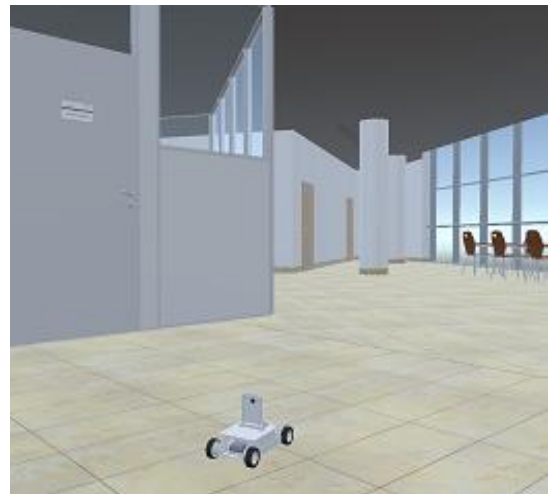
CECHY CHARAKTERYSTYCZNE ROZWIĄZANIA, KIERUNKI DALSZYCH PRAC:

Rozwiązanie pozwala na dwa sposoby pracy: uruchomienie symulatora na stacji roboczej i sterowanie za pomocą klawiatury lub joysticka (umożliwiając np. trenowanie przez naśladowanie operatora) oraz uruchomienie na serwerze i kontrolowanie symulacji za pomocą HTTP z możliwością równoległego uruchomienia kilku symulacji z różnymi ścieżkami.

TEAM PROJECT INFORMATION FOLDER – MAY 2024

DEPARTMENT OF ALGORITHMS AND SYSTEMS MODELLING

Project team: 4@KAMS'2023/24	1. Łukasz Nowakowski - leader 2. Konrad Bryłowski 3. Aleksander Czerwionka 4. Michał Krause 5. Krystian Nowakowski
Supervisor:	dr inż. Paweł Kowalski
Client:	dr inż. Krzysztof Manuszewski
Date:	26.05.2024
Key words:	autonomous vehicle, simulator, AI, unity



PROJECT TITLE:

Autonomous vehicle simulator

OBJECTIVES AND SCOPE:

The main objective of the project is to create an easy-to-use simulator which should help in training miniature vehicle AI without running it on the real model. The simulator will have two modes: following operator's example and self-learning, and two sceneries: follow the line type and ETI hall scene.

RESULTS:

After two semesters of project implementation, a simulator was developed in Unity featuring a random platform generator with a path and scenery of the first floor of the NE building. The simulator models a vehicle 3D-printed by the project supervisor, which behaves in a manner similar to the real vehicle under default simulation parameters. Data collection for drive force and wheel angle, as well as images of the model's surroundings, has been implemented. Additionally, a communication interface in the form of a web server with an API has been implemented, allowing for the initiation of new maps and vehicles on a map, as well as control of vehicle parameters: engine power and wheel angle. All these functions are also available through the user interface. According to the planned work organization, project progress is publicly available on the GitHub platform: <https://github.com/autonomous-vehicle-sim/Simulator>

MAIN FEATURES, FUTURE WORKS:

Final state of the project allows the user to control the simulator in two ways: running the simulation on a workstation and steering the vehicle with keyboard or joystick (allowing, for example, for training by following operator's example), and running the simulator on a server and controlling it via HTTP, with the capability of running several simulations in parallel with different paths.