



WYDZIAŁ ELEKTRONIKI,  
TELEKOMUNIKACJI  
I INFORMATYKI

## Dokumentacja Projektu grupowego

# Raport końcowy

Wydział Elektroniki, Telekomunikacji i Informatyki

Politechnika Gdańska

{wersja dokumentu wzorcowego: wersja 2/2023}

Nazwa i akronim projektu:	Zleceniodawca:	
Symulator pojazdu autonomicznego – SPA	dr inż. Krzysztof Manuszewski	
Numer zlecenia:	Kierownik projektu:	Opiekun projektu:
4@KAMS'2023/24	Łukasz Nowakowski	dr inż. Paweł Kowalski

Nazwa / kod dokumentu:	Nr wersji:
Raport końcowy – RK	1.00
Odpowiedzialny za dokument:	Data pierwszego sporządzenia:
Konrad Bryłowski	26.05.2024
	Data ostatniej aktualizacji:
	26.05.2024
	Semestr realizacji Projektu grupowego: 2

### Historia dokumentu

Wersja	Opis modyfikacji	Rozdział / strona	Autor modyfikacji	Data
1.00	wstępna wersja	całość	Konrad Bryłowski	26.05.2024

## Spis treści

1	Wprowadzenie - o dokumencie .....	3
1.1	Cel dokumentu .....	3
1.2	Zakres dokumentu .....	3
1.3	Odbiorcy .....	3
1.4	Terminologia .....	3
2	Rezultaty projektu .....	3
2.1	Wprowadzenie – opis ogólny projektu .....	3
2.2	Cel projektu i planowany zakres realizacji .....	4
2.3	Faktyczny zakres realizacji projektu i rozbieżności oraz zakres wykonanych prac .....	4
2.4	Osiągnięte wyniki .....	4
2.5	Charakterystyka pracy zespołowej .....	5
3	Załączniki .....	5

## 1 Wprowadzenie - o dokumencie

### 1.1 Cel dokumentu

Celem dokumentu jest zebranie istotnych informacji dotyczących całości zrealizowanego projektu w jednym miejscu i zaprezentowanie ich w przejrzysty sposób. Dokument ma na celu przede wszystkim ułatwienie oceny projektu, w szczególności w przypadku zgłoszenia go do nagrody lub wyróżnienia. Przewidywana objętość dokumentu 3-6 stron (części merytorycznej).

Należy również wskazać wykonane prace z podaniem ich krótkiej charakterystyki, wskazać rozbieżności wykonywanych prac w stosunku do planowanych, podsumować prace z wykazaniem pracy zespołowej oraz wyspecyfikować listę dokumentów wytworzonych w projekcie (wersji końcowych – lista obejmuje dokumenty z obu semestrów w wersjach ostatecznych), które zostały umieszczone i zatwierdzone przez opiekuna w serwisie SPG.

### 1.2 Zakres dokumentu

Dokument opisuje efekt prac nad projektem po dwóch semestrach realizacji, napotkane trudności i rozbieżności od zakładanego zakresu prac.

### 1.3 Odbiorcy

Odbiorcami dokumentu są członkowie zespołu projektowego oraz zleceniodawca i opiekun projektu, a także koordynator katedralny i wydziałowy.

### 1.4 Terminologia

API – interfejs aplikacji pozwalający na interakcję z nią poprzez systemy zewnętrzne

Framework – szkielet do budowy aplikacji, definiujący strukturę oraz ogólny mechanizm działania aplikacji

GitHub – platforma do przechowywania i publikowania projektów wykorzystujących system kontroli wersji Git

kanban – metoda planowania pracy zespołowej polegająca na umieszczaniu wszystkich zadań na tablicy z podziałem na kolumny np. „do zrobienia”, „w trakcie”, „do przetestowania”, „gotowe”

SI, AI – sztuczna inteligencja

UI – interfejs użytkownika

## 2 Rezultaty projektu

### 2.1 Wprowadzenie – opis ogólny projektu

Projekt polegał na opracowaniu symulatora pojazdu autonomicznego, którego celem jest pomoc w prowadzeniu zajęć z przedmiotu Architektury głębokiego uczenia oraz w prowadzeniu badań z obszaru sztucznej inteligencji. Symulator umożliwia trenowanie sztucznych sieci neuronowych do sterowania pojazdami autonomicznymi na podstawie obrazu kamer w symulacji zamiast z wykorzystaniem realnych pojazdów. Dodatkowym atutem takiego rozwiązania jest umieszczenie w symulacji wielu kopii tego samego pojazdu sterowanych za pomocą innych sieci neuronowych.

Zakładano dwa tryby pracy: trenowanie przez naśladowanie operatora sterującego pojazdem w symulacji oraz naukę samodzielną, a także dwa typy scenarii: podążanie za wytyczoną trasą (jak w konkursach UKMARS Line Follower i podobnych) oraz przemieszczanie się po realistycznej scenie (np. korytarz wydziału czy fragment kampusu).

System składa się z dwóch części: symulatora wykonanego na silniku Unity oraz serwera i aplikacji WWW.

Symulator zawiera zaplanowane dwa rodzaje scenarii oraz trybów sterowania, a także generator losowych scenarii typu line follower. Do realistycznej sceny wybrano zamodelowanie pierwszego piętra budynku NE.

Serwer i aplikacja WWW służą głównie do komunikacji z symulatorem. Dla użytkownika zapewniono interfejs umożliwiający zainicjowanie nowej symulacji, umieszczenie w niej pojazdu oraz sterowanie nim. Zostało zaprojektowane także API pozwalające sztucznej inteligencji na sterowanie pojazdami w symulacji oraz otrzymywanie obrazu z kamer.

Do działania wymagane jest uprzednie przygotowanie środowiska przez zainstalowanie Pythona oraz zależności projektu, które zgodnie z konwencją tego języka są umieszczone w odpowiednim pliku tekstowym.

## 2.2 Cel projektu i planowany zakres realizacji

Celem projektu jest opracowanie dedykowanego symulatora pojazdu autonomicznego do przeprowadzania badań z obszaru sztucznej inteligencji. Symulacja obejmuje pojazd wyposażony w kamery oraz otoczenie, w którym będzie się on poruszał. Głównym celem symulatora jest trenowanie sztucznych sieci neuronowych, które będą sterować pojazdami autonomicznymi na podstawie obrazu z kamer. Ponadto, symulator powinien umożliwiać umieszczenie w środowisku wielu pojazdów, każdy z nich sterowany za pomocą innej sieci neuronowej, co pozwala na badania różnych strategii sterowania.

Planowany zakres prac obejmował przegląd dostępnych na rynku rozwiązań, projekt symulatora, prace nad implementacją symulatora oraz fizyki symulacji oraz zaprojektowanie i implementacja interfejsu komunikacyjnego między sieciami neuronowymi a symulatorem.

Pierwotnie planowane były również badania zachowania kilku gotowych modeli sieci neuronowych, lecz wspólnie z opiekunem projektu podjęto decyzję o zastąpieniu ich dłuższym czasem na realizację prac implementacyjnych.

## 2.3 Faktyczny zakres realizacji projektu i rozbieżności oraz zakres wykonanych prac

Podczas realizacji projektu użyto Unity w wersji 2022.3.21f1 wraz z wbudowanym edytorem. Do pisania skryptów w C# używano środowiska programistycznego Visual Studio. Do tworzenia modelu pojazdu oraz sceny ETI użyto narzędzia Blender. Przy pracach nad serwerem i aplikacją WWW używano środowiska programistycznego PyCharm oraz Visual Studio Code. Aplikacja została wytworzona z wykorzystaniem frameworka Flask, biblioteki SQLAlchemy oraz bazy danych sqlite. Do komunikacji między symulatorem a aplikacją webową użyto WebSocket.

Prace były podzielone na kilka głównych komponentów:

- symulator w Unity – symulator implementujący fizykę jazdy zbliżoną do realnej, prace polegały na doborze odpowiednich parametrów fizyki symulacji oraz opracowaniu odpowiedniego modelu pojazdu
- generator map – generator wbudowany w symulator, prace polegały na wybraniu odpowiedniego algorytmu oraz jego implementacji w celu tworzenia ciągłej trasy w postaci krzywej na płaskiej powierzchni
- model pojazdu oraz pierwszego piętra budynku NE – modele wykorzystywane w symulatorze, prace polegały na wykonaniu zdjęć oraz pomiarów odległości, a następnie zamodelowaniu w Blenderze
- serwer i aplikacja WWW – aplikacja udostępniająca API oraz interfejs użytkownika, prace polegały na doborze frameworku, zaprojektowaniu API i bazy danych oraz implementacji
- komunikacja między serwerem a symulatorem – umożliwienie przekazywania poleceń otrzymywanych przez API lub UI do symulatora, prace polegały na doborze odpowiedniego interfejsu oraz implementacji po stronie serwera oraz symulatora

W pierwszym semestrze realizacji projektu planowano zamodelować fragment kampusu PG, jednak po konsultacji z opiekunem projektu zamodelowano korytarz pierwszego piętra NE z uwagi na rozmiary pojazdu dostarczonego przez opiekuna.

## 2.4 Osiągnięte wyniki

Opracowano symulator obsługujący dwa tryby: tryb manualnego sterowania pojazdem oraz tryb serwera, który pozwala na tworzenie wielu instancji map jak i pojazdów, które są sterowane zapytaniami wysyłanymi do serwera. Każdy z pojazdów posiada trzy kamery, z których zapis obrazu wraz z obecnymi parametrami pojazdu jest zapisywany na dysku.

Opracowany serwer oferuje następujące funkcjonalności: stworzenie nowej mapy, stworzenie nowego pojazdu o wybranych parametrach, usunięcie mapy lub pojazdu, ustawienie obrotów silnika wybranego pojazdu, ustawienie skrótu kół wybranego pojazdu, pobranie parametrów pojazdu lub zapisu z kamer danego pojazdu z dowolnego momentu symulacji, pobranie zdjęcia obrazującego wygląd danej mapy. Tworząc mapę, można wybrać jedną z dwóch opcji: stworzenie mapy z proceduralnie generowaną ścieżką z białej linii oraz stworzenie mapy odwzorowującej pierwsze piętro budynku NE. Opracowano również interfejs internetowy pozwalający na wygodne korzystanie z serwera.

Potencjalnymi ścieżkami rozwoju może być dodanie do projektu wbudowanej sztucznej inteligencji która by samodzielnie mogła wytrenować model pozwalający na jazdę przy pomocy oferowanego interfejsu, potencjalne jeszcze lepsze odwzorowanie modelu jazdy samochodu w prawdziwym życiu oraz udostępnić więcej opcji konfiguracji parametrów pojazdów.

## 2.5 Charakterystyka pracy zespołowej

Zespół przyjął organizację pracy według metodyki Kanban dostosowanej do potrzeb i natury pracy nad projektem studenckim. Spotkania zespołu odbywały się raz w tygodniu, a spotkania z opiekunem odbywały się nieregularnie, w zależności od potrzeb bieżących stacjonarnie na uczelni lub przez Microsoft Teams. Podczas spotkań członkowie zespołu sami przydzielali sobie zadania oraz zgłaszali nowe. Komunikacja z opiekunem w sprawach bieżących odbywała się za pośrednictwem kierownika projektu. Do współdzielenia kodu założono repozytorium w serwisie GitHub.

Poszczególni członkowie zespołu wykonywali następujące prace:

- Konrad Bryłowski – koordynacja pracy nad dokumentacją, wybór frameworka do serwera i aplikacji WWW, zaprojektowanie i implementacja API, implementacja komunikacji między symulatorem a serwerem
- Aleksander Czerwionka – przegląd i prezentacja dostępnych silników, przygotowanie podstawowej sceny symulacji, generator map, implementacja komunikacji między symulatorem a serwerem, utworzenie skryptu do zapisu obrazu kamer oraz dostosowanie symulacji do obsługi wielu pojazdów jednocześnie
- Michał Krause – interfejs użytkownika symulatora oraz aplikacji webowej, modelowanie pojazdu oraz korytarza ETI, prace nad fizyką symulacji
- Krystian Nowakowski – przegląd i prezentacja dostępnych rozwiązań dotyczących fizyki, dostosowanie fizyki tak, aby pojazdy na tej samej mapie nie wpływały na siebie wzajemnie, dodanie opcji zapisania widoku mapy z góry na potrzeby interfejsu użytkownika
- Łukasz Nowakowski – kierownik projektu, interfejs użytkownika aplikacji webowej, prace implementacyjne w aplikacji webowej, wykonywanie zdjęć na potrzeby modelowania korytarza ETI

## 3 Załączniki

*{dotyczy wszystkich dokumentów z SPG - pełna lista aktualnych-końcowych wersji dokumentów, wypracowanych w trakcie realizacji projektu przez 2 semestry, wraz z wyszczególnioną aktualną wersją raportu końcowego; w szczególności:*

*- umieszczamy tutaj listę dokumentów z 1 semestru (zgodną z treścią raportu semestralnego), ale w najnowszych wersjach (jeśli podlegały korekcie, rozszerzeniu itp. i oznaczono je nowymi numerami wersji)*

*- umieszczamy tutaj listę dokumentów z 2 semestru w najnowszych wersjach, w tym również wpisujemy raport końcowy}*

Tabela. 3.1. Specyfikacja opracowanych dokumentów

L.p.	Nazwa dokumentu	Nazwa pliku umieszczonego w SPG