



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Dokumentacja Projektu grupowego
Dokumentacja techniczna projektu
Wydział Elektroniki, Telekomunikacji i Informatyki
Politechnika Gdańska

{wersja dokumentu wzorcowego: wersja 2/2023}

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

Nazwa / kod dokumentu: Dokumentacja techniczna produktu – DTP	Nr wersji: 2.00
Odpowiedzialny za dokument: Konrad Bryłowski	Data pierwszego sporządzenia: 20.11.2023
	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	14.11.2023
1.01	ujednoczenie informacji z opublikowanymi w SPG	str. 1	Konrad Bryłowski	20.01.2024
1.02	uwzględnienie nowych funkcji	rozdział 2	Konrad Bryłowski	24.01.2024
2.00	uwzględnienie funkcji dodanych podczas pracy na drugim semestrze	całość	Michał Krause	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	Dokumentacja techniczna projektu	3
3	Załączniki	4

1 Wprowadzenie - o dokumencie

1.1 Cel dokumentu

Celem dokumentu jest udokumentowanie informacji dotyczących produktu, jego cech funkcjonalnych, parametrów technicznych, schematów blokowych, oprogramowania, wyników działania, zdjęć produktu, pomiarów, testów oraz innych elementów wymaganych przez opiekuna i klienta.

1.2 Zakres dokumentu

Dokument obejmuje podjęte przez zespół decyzje dotyczące projektu symulatora oraz dotychczasową ich realizację.

1.3 Odbiorcy

Odbiorcami dokumentu są członkowie zespołu projektowego oraz zleceniodawca i opiekun projektu.

1.4 Terminologia

assets – reprezentacja rzeczywistych obiektów w silniku symulacji (np. graficzny model budynku, dźwięk przejeżdżającego samochodu, animacja)

Blender – narzędzie do modelowania i renderowania obraz

convex hull – najmniejszy zbiór wypukły zawierający dany podzbiór

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

prefab – inaczej prefabrykat; zasób, który pozwala przechowywać obiekty wraz z ich komponentami i innymi danymi w konfigurowalnych szablonach

silnik – framework zawierający konieczne biblioteki oraz środowisko m. in. do edytowania scenów trójwymiarowych

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

2 Dokumentacja techniczna projektu

2.1 Omówienie

Projekt został podzielony na kilka współpracujących komponentów. Pierwszym z nich jest symulator działający na silniku Unity. Pozwala on na samodzielne sterowanie pojazdem, bądź też uruchomienie klienta, który jest w stanie obsłużyć wiele pojazdów jednocześnie poprzez obsługę API. Symulator obsługuje proceduralnie stworzone mapy, jak i jazdę po zamodelowanym korytarzu na pierwszym piętrze wydziału ETI Politechniki Gdańskiej. Podczas jazdy ręcznej, jak i poprzez API zbierane są dane z jazdy - obrazy z kamer oraz parametry jazdy. W taki sposób zebrane dane mogą zostać wykorzystane do trenowania sztucznej inteligencji do jazdy rzeczywistym pojazdem.

Drugim komponentem projektu jest serwer, który służy do komunikacji z symulatorem poprzez API. Obsługuje on tworzenie nowych symulacji, pojazdów, tras oraz sterowanie stworzonymi pojazdami. Zwraca on dane uzyskane z jazdy, czyli obrazy z kamer i parametry jazdy.

Ponadto serwer udostępnia interfejs sieciowy, w którym można w prosty sposób korzystać z API symulatora. Umożliwia stworzenie symulacji, dodanie do niej pojazdów z wybranymi parametrami, jak i wejście w szczegóły jazdy każdego z pojazdów.

2.2 Funkcje symulatora

2.2.1 Zbieranie danych

W czasie jazdy symulator zapisuje parametry jazdy (moc silnika, skręt kół) oraz obrazy otoczenia pozyskane z kamer rozstawionych na pojeździe. Uzyskane dane można wykorzystać do uczenia sieci neuronowych rzeczywistych pojazdów w taki sam sposób, jak z danych pozyskanych w prawdziwym otoczeniu.

2.2.2 Generator tras

Jednym z głównych przypadków użycia symulatora jest nauczanie pojazdu do jazdy tak, aby nie zbacał z linii białej narysowanej na czarnej trasie. Plansza taka często spotykana jest na zawodach, które właśnie polegają na jak najlepszym przejechaniu trasy przez pojazd.

Generator losowych tras generuje zbiór punktów na mapie, z których przy pomocy algorytmu Convex Hull wybiera punkty, z których będzie utworzona trasa. Następnie algorytm sprawdza czy wybrane punkty nie są zbyt blisko siebie i jeśli są to zmienia iteracyjnie ich położenie względem siebie.

Następnie algorytm interpoluje ścieżkę między tymi punktami za pomocą krzywych Béziera drugiego stopnia.

2.2.3 API symulatora

Symulator wyposażony został w interfejs API, który pozwala na korzystanie z symulatora poprzez sieć. Na system ten składają się dwa elementy – klient symulatora działający na silniku Unity, który obsługuje tworzenie symulacji, pojazdów i tras. Drugim komponentem jest serwer napisany w języku Python, przy użyciu biblioteki Flask, działający na protokole HTTP. Udostępnia on liczne punkty końcowe, dzięki którym można tworzyć i usuwać obiekty w symulatorze, jak i otrzymywać dane z symulatora. Komunikacja między tymi elementami odbywa się za pomocą mechanizmu Websocket.

Serwer korzysta z bazy danych SQLite, zarządzanej poprzez moduł SQLAlchemy do zapisywania danych symulatora w celu przesyłania ich do użytkowników. Danymi tymi są informacje o danej symulacji oraz o konkretnym pojeździe.

Serwer udostępnia dokumentację swoich punktów końcowych pod adresem “/api”, gdzie dokładnie opisana jest każda funkcja dostępna z poziomu API. Dokumentacja ta została wykorzystana przy pomocy narzędzia Swagger.

2.2.4 Interfejs sieciowy symulatora

Oprócz API symulatora, serwer również udostępnia łatwy w obsłudze interfejs sieciowy, który komunikuje się z resztą części projektu. Można w nim utworzyć nową symulację, zobaczyć listę wszystkich symulacji, stworzyć w danej symulacji pojazd oraz otrzymać z danego pojazdu informacje oraz obraz z kamer z jazdy. Interfejs ten napisany jest w postaci plików szablonów Jinja modułu Flask serwera.

2.2.5 Jazda po holu pierwszego piętra nowego budynku WETI

Podczas wyboru trasy, na której umieszczamy pojazd, mamy możliwość jazdy po zamodelowanym pierwszym piętrze nowego budynku wydziału ETI na Politechnice Gdańskiej. Model ten został stworzony w narzędziu Blender.

3 Załączniki

brak załączników