G-One Debuggers

Moway

Software Requirements Specification

For Virtual self-driving car using reinforcement learning

Version 1.0

V.Sajeevan

160544C

15th February, 2019

Revision History

Date	Version	Description	Author
15/02/2019	1.0	Initial Requirements Specification document	`V.Sajeeavn

Table of Contents

Software Requirements Specification

1. Introduction

The introduction of the Software Requirement Specification (SRS) provides an overview of the entire SRS. It includes the purpose, scope, definitions, acronyms, abbreviations, references and the overview of the SRS. This document explains the whole software scenario for Moway, Virtual self-driving car using reinforcement learning and user by defining software requirements for functionalities and non- functionalities. Moreover, the technologies used, software design constraints and assumptions for the above topics are also included in this document.

1.1. Purpose

The purpose of this SRS document is to analyze all the functional and non-functional requirements that define the system for this Virtual self-driving car simulator. For further development, more requirements, technologies, and software design architecture are sorted out so that goals which should be achieved during the system development will be identified easily.

1.2. Scope

The scope of this project is to develop a framework, Virtual self-driving car can be simulatable in different traffic situations like driving on the straight road, curved road, junction, and traffic with manned vehicles.

1.3. Definitions, Acronyms, and Abbreviations

Virtual self-driving car	A self-driving car is a vehicle that uses a combination of sensors, and artificial intelligence to travel between destinations without a human operator. Not physically existing as such but made by software to appear to do so.
Rule-based car or manned vehicle	Rule-based cars are used as a way to store and manipulate knowledge to interpret information in a useful way. They are often used in artificial intelligence applications and research
UI	User interface

1.4. References

[1] PAN, X., YOU, Y., WANG, Z. AND LU, C.
Virtual to Real Reinforcement Learning for Autonomous Driving
In-text: (Pan et al., 2019)

[2] Journal STILGOE, J.
Machine learning, social learning and the governance of self-driving cars
In-text: (Stilgoe, 2017)

1.5. Overview

The overall description explains the overview of the system. The basic functionalities of the system will be described with the general matters regarding the system and the requirements without stating the specific requirements. Moreover, the background of the requirements and such details are also discussed here. Finally, the assumptions and constraints regarding the project are described. Also, the technologies which have been planned to be used are also provided in this document.

2. Overall Description

2.1. Product Perspective

There is more type of autonomous vehicles in this new technology. But there are only a few simulating frameworks to test those autonomous vehicles and lack of tools to simulate different traffic phenomena in the presence of autonomous cars.

The perspective of this product makes the testing of autonomous vehicles easier in virtual. This framework can be simulatable in different traffic situations like driving on the straight road, curved road, junction, and traffic with manned vehicles. Users can change the map path and they can add the obstacles and manned vehicles along with the virtual self-driving car.

2.2. Product Functions

This product is a desktop offline framework it can simulatable an autonomous vehicle in different phenomena. The autonomous vehicle should be able to analyze the environment and store the data in local storage. Able to give predictions about the environment. The environment should be able to change the user's likeness

2.3. Product Characteristics

Operators should have good experience in computer literacy and knowledge regarding the autonomous vehicle. They are expected to have just the basic technical knowledge to use the computer to simulate the framework. The system will be an offline system with a user-friendly interface to meet those requirements.

2.4. Constraints

The system will be run on Microsoft Windows Vista / Windows 7 with Pentium 4 or higher. Python will be the implementation language in the JetBrains PyCharm IDE platform. The database is handled by NumPy python library it will be stored in local storage.

3. Specific Requirements

This section states all the requirements required in virtual self-driving car and simulation framework. All the requirements are stated with details and they are divided into categories depending on their area

3.1. Functionality

3.1.1. Store trained data

Numpy Library is using as a database handle platform for track trained data. Training time is just too big. So it must be saved in anywhere. This framework can be storable or loadable in local storage for a particular virtual self-driving car.

3.1.2 Add / remove obstacles

The users want to simulate in a different environment and different phenomena. Such that user can add or remove the obstacles for creating a different environment.

3.1.3 Add Manned vehicles

The users want to simulate in a different environment and different phenomena. Such that user can add or remove the Manned vehicles for creating different phenomena.

3.1.4 Choose the map

The user should be able to create an environment by choosing different maps.

3.1.5 Simulate in different phenomena

The user can train or test the autonomous vehicle after creating a different environment while using these functionalities.

3.2 Usability

3.2.1 Training time for the users

This application should not be very complicated. Usually, autonomous systems need high time duration to train. In this framework, the user can save the trained data in local storage so the user does not want to re-train the vehicle.

3.2.2 Friendly and attractive user interface

Users of the system should be able to interact with the system easily and the interaction should be easily understandable without any uncertainties.

3.2.3 Ease to use

The users will not need any extra skills or high technical knowledge to use the system. Basic knowledge about using the computer will be enough to use the system.

3.3 Reliability

3.3.1 Availability

This framework is an offline desktop framework.it also available at any time for the user unless there are any updates or problems.

3.3.2 Accuracy

Automated systems analyzing and forecasting should be very accurate. Forecasting error should be minimum. The forecasted error should be varied within a 5% range from the actual value. The system will not lose any trained data files from the framework and will make sure that the files are saved in local storage.

3.4 Performance and Security

Security is less priority for this framework. In performance wise, the framework should be user-friendly, appropriate error messages should be displayed and the system should be handling several inputs simultaneously. The system is expected to be responsive all the time with minimum response time.

Since the training through the dataset is computationally exhaustive at times the training gets interrupted and hence it slows down the development of the system. So to mitigate it exploit the different system to develop and simulate.

3.5 Supportability

The naming of the classes and the tables of the database is expected to be simple and make sense. This is because there is the possibility of the security measures being changed with time and the system will have to be changed accordingly as well. The code is expected to be commented where necessary thus facilitating easy understanding. The program logic is expected to be simple to avoid unnecessary complexities arriving at any failure.

3.6 Design Constraints

3.6.1 Software languages

This framework will be designed using Python and some of the Python Libraries. those are Pygame, Pymunk Libraries are used for UI development, Numpy Libray used to manage the database of autonomous vehicles Which are the popular and highly used languages and Libraries to develop desktop applications.

3.6.2 Development tools

The system will be using Python Numpy library as the database. JetBrains PyCharm IDE will be used to develop the system.

3.7 Purchased Components

There are no purchased components in this framework.

3.8 Interfaces

3.8.1 User Interfaces

- Choose map interface

  The users want to simulate in a different environment and different phenomena. Such that the user should be able to create an environment by choosing different maps.

- Add obstacles and manned vehicles interface

  The users want to simulate in a different environment and different phenomena. Such that user can add or remove the obstacles or add manned vehicles for creating a different environment and different phenomena.

- Simulating interface

  The user can train or test the autonomous vehicle after creating a different environment while using these interface.

3.8.2 Hardware Interfaces

The Moway framework doesn't require any special hardware devices. Users can use the system using their existing devices such as Desktops and Laptops.

3.8.3 Software Interface

The framework does not need any specific software interface for the client side usage. Users can use the platform using their existing operating system.

3.8.4 Communications Interfaces

Automated vehicles need to communicate with the local databases which are stored in local storage

3.9 Database Requirements

The Moway framework will be using Numpy python library database as the centralized data storage for store autonomous vehicle's trained data. It is a free and open-source document-oriented library. Those data will store on a local disk. The user can Load the trained data for simulating the same system.

4. Supporting Information

- Table of contents is given on page 3 of the document.
- System and all other components are developing using Python. Microsoft Windows Operating Systems. And all the more advanced Windows operating systems.