ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Project Report
Version 1

CENG 407
Innovative System Design and Development I

202113
Automated Self Learning Bus Simulation

Metehan KOÇ
201711044
Mustafa Timur BEŞLİ
201811013
Okan YILDIZ
201811068

Dr. Gül TOKDEMİR

Table of Contents

Abstract

Machine learning is gaining importance in areas where systems need to accurately learn and adapt to new patterns with minimal human intervention. One such area is the rise of autonomous cars, with many big car manufacturers trying to improve their self-driving automobiles such as Tesla, Ford and Audi. The speed of development in this area is turning the idea of self-driving automobiles into reality, with the hope of eradicating non autonomous cars in cities which would drastically reduce car accidents and traffic, this also helps cars run more efficient lessening their effect on environment. Our project aims to see how a self-driving bus may be implemented using machine learning, the bus must be able to drive on a predetermined route within the boundaries of the road. We will simulate the bus and the road to test how our algorithm adapts to the route, after an acceptable result we will add obstacles for the bus to adapt once more.

Öz

Makine öğrenimi sistemlerin yeni durumlara asgari insan yardımı ile kesin bir şekilde adapte olması gereken alanlarda önem kazanıyor. Bu alanlardan biri ise kendi kendine süren arabalar Tesla, Ford ve Audi gibi büyük otomobil üreten firmalar kendi kendini süren arabalarını geliştirmeye çalışıyor. Bu alandaki gelişmenin hızı kendi kendini süren araçları fikirden gerçeğe çeviriyor, hedefleri ise şehirlerde kendi kendine süremeyen arabalardan kurtulmak, bu da şehirlerde kazaları ve trafiği yok denecek kadar azaltır, ayrıca arabaların daha verimli çalışmasını ve çevreye verdikleri zararı da azaltır. Bizim projemiz makine öğrenmesi kullanarak kendi kendini süren bir otobüs yapmak, otobüs yol sınırlarına uyarak önceden belirlenmiş rotasında gitmeyi öğrenecek. Otobüsü ve yolu simüle edeceğiz ve algoritmamızın nasıl yola adapte olduğunu test edeceğiz, kabul edilebilir bir sonuç aldıktan sonra ise otobüsün rotasına engeller koyup yeniden adapte olmasını bekleyeceğiz.

1. Literature Review

1.1 Introduction

Fully automated buses are expected to improve the competitiveness of public transportation services by eliminating the cost and performance constraints of human drivers and significantly reducing traffic accidents, thereby potentially revolutionizing the existing public transportation system. In an environment where centralized information is not available, multi-agent control systems are important and require agents to collaborate with other agents because they may not contain all the data or resources needed to achieve the goal. Self-driving buses must learn to travel effectively in cities in order to integrate into the public transportation system. This can be achieved through reinforcement learning with intelligent agents. In this study, we will simulate the problem of automatic bus fleet control, which is a hard task for the agents due to random arrivals and incomplete observations of the environment. In response to this problem, we will be using a multi-agent reinforcement learning method . An agent-based simulation platform will be developed in the UNITY game engine to model the dynamic system of fixed stops/station loops, automated bus fleets, and passenger. We will be using The Unity Machine Learning Agents Toolkit (ML-Agents) which is an open-source project that enables games and simulations to serve as environments for training intelligent agents. Our goal is to make a public bus driven by artificial intelligence that can reach all the bus stops in a 3D city on time and pick up and drop off passengers.

Objectives

- The bus will be able to adapt to any particular city and bus stop patterns.
- The bus will follow the lanes on the road and drive in accordance with the traffic rules.
- In city simulation there will be cars moving in traffic, pedestrians walking on the pavement, and crossing in a three-dimensional observable way.
- When we change the city in the simulation environment (the locations of the bus stops, the road map of the city, the flow of traffic, etc.) The bus will be able to quickly adapt to all these changes.

1.2 Previous Work Done

Self-Parking Car Simulation using Reinforcement Learning Approach for Moderate Complexity Parking Scenario

Autonomous parking system is the best way for reducing time waiting to park in the parking spaces or looking for a parking space, particularly in heavily urbanised areas. We propose the auto-parking car simulation framework using proximal policy optimization for deep reinforcement learning in a moderately complex parking scenario which consists of basic and difficult zones for parking. Different configurations are explored including sparse and dense rewards combined with checkpoints, orientation rewards and stay-in-collision punishment. It shows high success parking rate in a basic parking zone up to at more than 95%. For the more challenging parking zones, the model works better when each zone is trained individually.[3]

Deep Learning Algorithm using Virtual Environment Data for Self-driving Car

Recent exceptional progresses in artificial intelligence researches enable many attempts to implement self-driving cars. However, in real world, there are a lot of complicated risks and costly problems to acquire training data for self-driving artificial intelligence algorithms. This paper introduces an algorithm to collect training data from a driving game, which highly simulates the environment of the real world. In the data collection part, the algorithm gathers both driving game screen image and control value. We utilize the collected data from virtual game environment to learn a deep neural network. Experimental result for applying the virtual driving game data to guide a children's car demonstrates the effectiveness of the proposed algorithm.[4]

1.3 Sample Scenario on Risks

As great as it sounds, vehicles like self-driving cars, buses, or even airplanes aren't completely foolproof. There are potential risks and challenges in Automated Self-Learning Bus Simulation that may arise from the environment, people, or the system. To make these problems more understandable, we can go through certain headings.

Systemic Challenges

First of all, we must develop the system we have built as much as possible before introducing environmental factors. In order to develop the system, we need a large amount of processing power. As the CPU and GPU components of the systems we use as a team are more suitable for daily use, we will be able to use machine learning models more limitedly. If we had professional equipment, we would have more modeling and testing opportunities. A 2020 AAA study[1] shows that despite the vast opportunities large corporations have, they found that, on average, during 4,000 miles of real-world driving equipped with active driver assistance systems, they experience some type of problem every 8 miles.

Real Life Driving Conditions

While the Automated Self-Learning Bus is technologically equipped to help prevent human error and reduce traffic accident rates, the cities we live in may not be ready for the Automated Self-Learning Bus. Erased lanes, narrow streets, illegal actions of people we may encounter in traffic, cars parked in front of stops or obstacles placed in front of the stops, etc. everyday problems can cause the Automated Self-Learning Bus to give unexpected errors.

Ethical Concerns

One of the phenomena that come to mind when talking about a self-driving vehicle is what the bus, which moves within the speed limits without any traffic violations, will do in case of a person or animal suddenly jumping onto the road. Should he crush any animal or human that he encounters? Or should he go off the road and risk the lives of those on the bus? In a survey conducted in the USA[2], the majority of the participants preferred the vehicle to have an accident rather than crushing the pedestrian.

Cyber Attacks

No matter how advanced the systems (firewall) of the Automatic Self-Learning Bus are against possible cyber attacks, it is not 100% inaccessible from the outside like any device connected to the internet. In a possible scenario, if the attacker is targeted by individuals or groups, the lives of the people on the bus and the people in the vehicles sharing the same road with the bus may be endangered.

1.4 Conclusion

As a result, in many parts of the world, we have seen that large car or technology companies or small venture groups are working on self-driving vehicles, and even a few large car companies are quite successful. However, when it comes to self-driving vehicles, the first thing that comes to mind is cars or long-distance trucks, and we have seen that not that much work has been done on buses. We got an idea about the previous self-driving vehicle studies and the problems that may be encountered. Our Automated Self-Learning Bus is aimed at cities with less traffic, fewer accident rates, and are more environmentally friendly

1.5 References

1 - Ellen Edmonds Manager. (2020, December 10). *AAA finds active driving assistance systems do less to assist drivers and more to interfere*. AAA Newsroom. Retrieved November 12, 2021, from https://newsroom.aaa.com/2020/08/aaa-finds-active-driving-assistance-systems-do-less-to-assist-drivers-and-more-to-interfere/.

2 - *The Social Dilemma of Autonomous Vehicles*. ResearchGate. (n.d.). Retrieved November 12, 2021, from https://www.researchgate.net/publication/301293464_The_Social_Dilemma_of_Autonomous_Vehicles.

3 - B. Thunyapoo, C. Ratchadakorntham, P. Siricharoen and W. Susutti, "Self-Parking Car Simulation using Reinforcement Learning Approach for Moderate Complexity Parking Scenario," 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology.
https://ieeexplore.ieee.org/document/9158298

4 - J. Kim, G. Lim, Y. Kim, B. Kim and C. Bae, "Deep Learning Algorithm using Virtual Environment Data for Self-driving Car," 2019 International Conference on Artificial Intelligence in Information and Communication. https://ieeexplore.ieee.org/document/8669037

2. Project Work Plan

Project Work Plan

Start Week: Eki 11, 2021

Advisor: Gül Tokdemir

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Starting	Eki 11	Eki 18	Eki 25	Kas 1	Kas 8	Kas 15	Kas 22	Kas 29	Ara 6	Ara 13	Ara 20	Ara 27	Oca 3	Oca 10	Oca 17	Oca 24
Team Setup	TEAM															
Project Proposal Form		TEAM														
Project Selection Form		TEAM														
Project Work Plan			TEAM	TEAM												
Literature Review			TEAM	TEAM	TEAM											
Software Requirements Specification						TEAM	TEAM	TEAM	TEAM							
Project Webpage								TEAM	TEAM	TEAM						
Software Design Description										TEAM	TEAM	TEAM				
Project Report													TEAM	TEAM		
Project Tracking Form													TEAM	TEAM		
Presentation															TEAM	TEAM

3. Software Requirements Specification

3.1 Introduction

Purpose

The purpose of this document is to outline the requirements for the Autonomous Bus Simulation project. It will describe features, parameters and goals. Also it will lay out how we plan to develop the project.

Scope

This project aims to create a self-driving bus that learns and adapts to the road conditions. The vehicle will examine its environment and act accordingly. The algorithm will be given various data through the sensors on the bus to reach the correct action. Our environments will be generated using the Unity Engine. We will be using Machine Learning concepts to create the algorithm.

Glossary

Terminology	Definition
Passenger	Person or persons using the Automated Self-Learning Bus to travel between specific bus stops.
Bus Stop	Predetermined places where passengers can perform their get-off and get-on functions.
Autonomous Driving	An autonomous car is a vehicle capable of sensing its environment and operating without human involvement.[1]
SRS	Software Requirements Specification
ML	Machine Learning
CPU	Central Process Unit

Overview

The remaining sections of this document provides a general description. Overall description of this project is discussed in section 2. Section 3 includes the functional requirements and details of the requirements for developers. Section 4 includes the various References.

3.2 Overall Description

Product Perspective

First of all we will create the environment for the bus to learn, we will create a path with several obstacles and stops. The bus will be implemented with several sensors to calculate its distance between obstacles, stops and edges of the road. After we have the data necessary we will implement a Machine Learning algorithm to choose which action the bus should take.

3.3 Requirement Specification

3.3.1. External Interface Requirements

3.3.1.1. User Interfaces

The user starts the system and determines the locations of the bus stops on the map and the order in which they will be followed.

3.3.1.2. Hardware Interfaces

Currently uses the environmental sensors and cameras in the Simulation. If it will be used in real life in the future, it needs advanced sensors such as LIDAR[2] and a control unit to process the data from these sensors.

3.3.1.3. Software Interfaces

The simulation environment requires Unity Engine. Python compiler is required to use machine learning algorithms.

3.3.1.4. Communication Interfaces

There are no external communications interface requirements.

3.3.2. Functional Requirements

3.3.2.1. Use Cases

Use Case	**Description**
Start Program	User starts the simulation.
End Program	User ends simulation.
Environment Controller	Checks sensors to estimate distance with the environment.
Brake	Slows down the bus.
Accelerate	Increases the speed of the bus.
Stop	Bus breaks until speed is 0.
Red Light	Traffic light with red light on.
Green Light	Traffic light with green light on.
Bus Stop	Bus stop in sensor range.
Road Outline	Guides bus to stay within the road limits.

3.3.2.1.1 Case: Start Program

Overview

User starts the simulation.

Response

The road is generated, bus starts its path, the sensors start and continues until simulation is ended.

3.3.2.1.2 Case: End Program

Overview

User ends the simulation.

Response

Simulation is ended, total distance travelled and number of bus stops achieved is given to user.

3.3.2.1.3 Case: Enviroment Controller

Overview

Sensors on the bus returns distance of objects that are in sensor range.

Response

Sensor data is given to the machine learning algorithm to decide the next response.

3.3.2.1.4 Case: Brake

Overview

Slows the bus down when an obstacle or a red light appears.

Response

When sensors detect an obstacle or a red light the bus starts slowing down to increase its time to reach the object.

3.3.2.1.5 Case: Accelerate

Overview

Increases the buses speed when sensors do not detect an obstacle.

Response

The bus starts to accelerate when the traffic light turns green or when an obstacle disapears from sensor range.

3.3.2.1.6 Case: Stop

Overview

Slows the bus down to stop.

Response

If the bus gets close to obstacle or a red light despite slowing down it is stopped until the reason is cleared.

3.3.2.1.7 Case: Red Light

Overview

Sensors detect a red light in front of the bus ands slows down.

Response

Slows the bus down when a red trafic light is seen in front of the bus and stops if the bus gets too close.

3.3.2.1.8 Case: Green Light

Overview

Sensors detect a greenlight in front of te bus.

Response

Increases the buses speed when an obstacle is cleared or the red light turns green.

3.3.2.1.9 Case: Bus Stop

Overview

Sensor detects a bus stop and changes route to it.

Response

Sensor detects a bus stop and changes route to stop on it.

3.3.2.1.10 Case: Road Outline

Overview

Sensors measure the distance to road boundries.

Response

Sensors measure the distance to road boundries andsteers away if too close.

3.4 Non-functional Requirements

3.4.1. Performance Requirements

- The Automated Self-Learning Bus must quickly analyze and react to environmental changes while in motion. Response time should be minimal.
- It should determine the route to be taken in the shortest time between bus stops in the most effective way. In the face of unexpected situations in traffic, he should review new routes and reach the bus stop within the expected time interval.

3.4.2. Safety Requirements

- The Automated Self-Learning Bus must act within the traffic rules and avoid accidents that could endanger the life of the passengers in extraordinary situations.
- The Automated Self-Learning Bus should analyze the environmental elements well and avoid the mistakes of careless people or other creatures. For example, a distracted person suddenly jumping onto the road, children running on the streets open to traffic, or street animals crossing the street.

3.4.3. Security Requirements

- The Automated Self-Learning Bus must have a defense system against cyber attacks. Potential cyber attacks of individuals or groups with bad intentions may endanger the lives of passengers or pedestrians or cause accidents with material damage.
- In another possibility, malicious attackers can take the bus off its route for fun, preventing it from working properly. For example, the city bus travels between cities after the attack.

3.5 References

[1] - *What is an autonomous car?* Synopsys. (n.d.). Retrieved December 8, 2021, from https://www.synopsys.com/automotive/what-is-autonomous-car.html.

[2] - US Department of Commerce, N. O. and A. A. (2012, October 1). *Lidar*. NOAA's National Ocean Service. Retrieved December 8, 2021, from https://oceanservice.noaa.gov/facts/lidar.html.

4. Software Design Document

4.1 Introduction

This document provides detailed information about Automated Self Learning Bus Simulation system software. It will explain how the proposed method works and The user interface of this method. The purpose of this document is to provide guidance for users.

4.1.1 Purpose

The purpose of this document is to provide detailed information on the proposed project "Automatic self-learning bus simulation". This document contains the architecture design description of the Automated Self Learning Bus Simulation. It includes the features, specifications, functionalities of the system. It shows how the use cases shown in the SRS are implemented in the system.

4.1.2 Scope of Project

This project aims to create a self-driving bus that learns and adapts to the road conditions. The vehicle will examine its environment and act accordingly. The algorithm will be given various data through the sensors on the bus to reach the correct action. Our environments will be generated using the Unity Engine. We will be using Machine Learning concepts to create the algorithm.

4.1.3 Glossary

Terminology	Definition
Passenger	Person or persons using the Automated Self-Learning Bus to travel between specific bus stops.
Bus Stop	Predetermined places where passengers can perform their get-off and get-on functions.
Autonomous Driving	An autonomous car is a vehicle capable of sensing its environment and operating without human involvement.[1]
SRS	Software Requirements Specification
ML	Machine Learning
CPU	Central Process Unit

4.2 System Architecture

4.2.1 Description of Problem

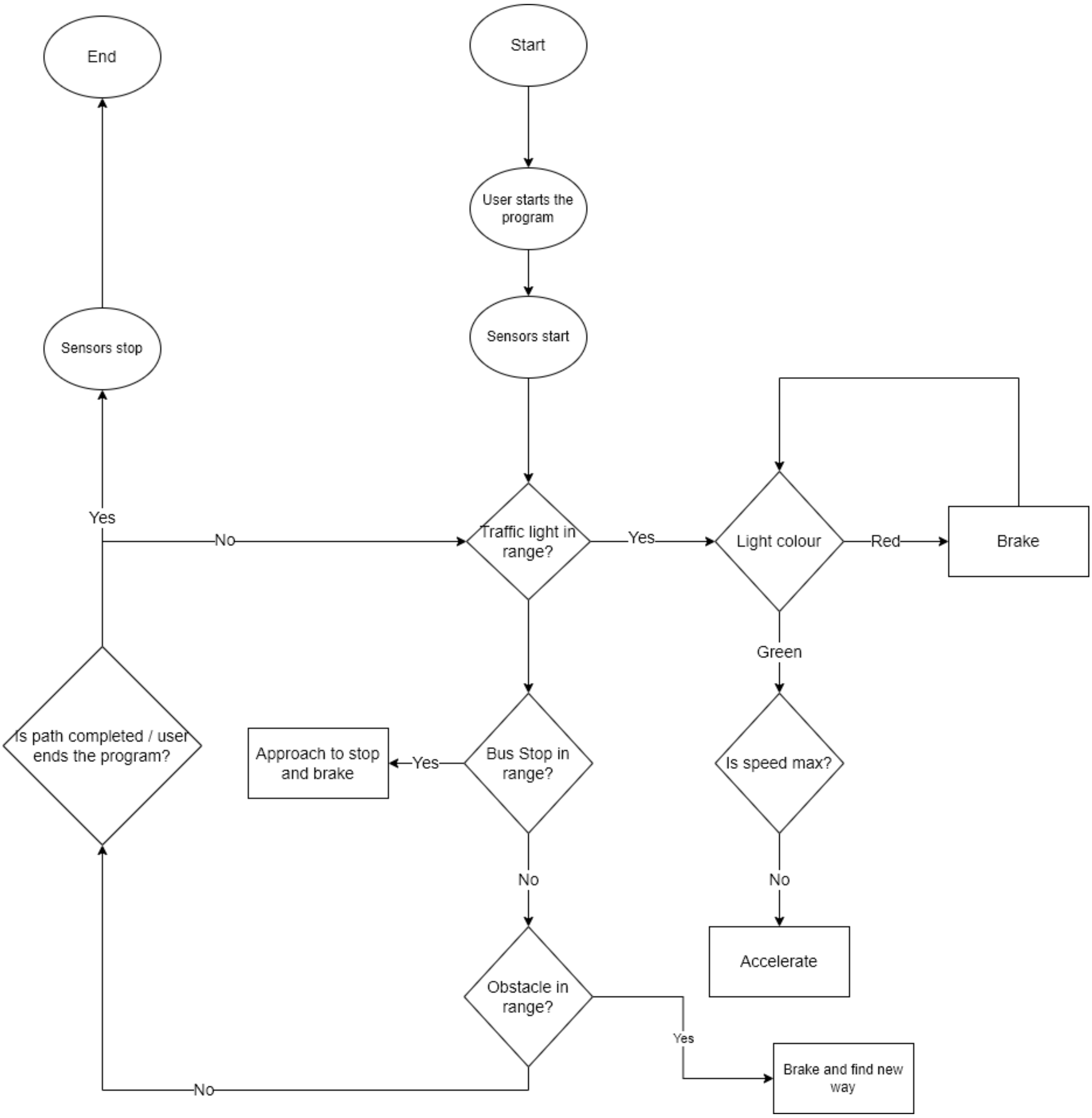
The average person spends 70 hours a year in traffic. We aim to prevent this, to save time for people and to cause less harm to the environment with minimum fuel consumption. In addition, we aim to prevent human errors and increase the safety of our passengers. If what we do in simulation is successful enough, it is very likely that we will see the results of this experience in the real world.

4.2.2 Technologies/Tools Used

This simulation will be built on Unity Engine using Unity Engine ML Libraries[1]. C# language will be used as the programming language.

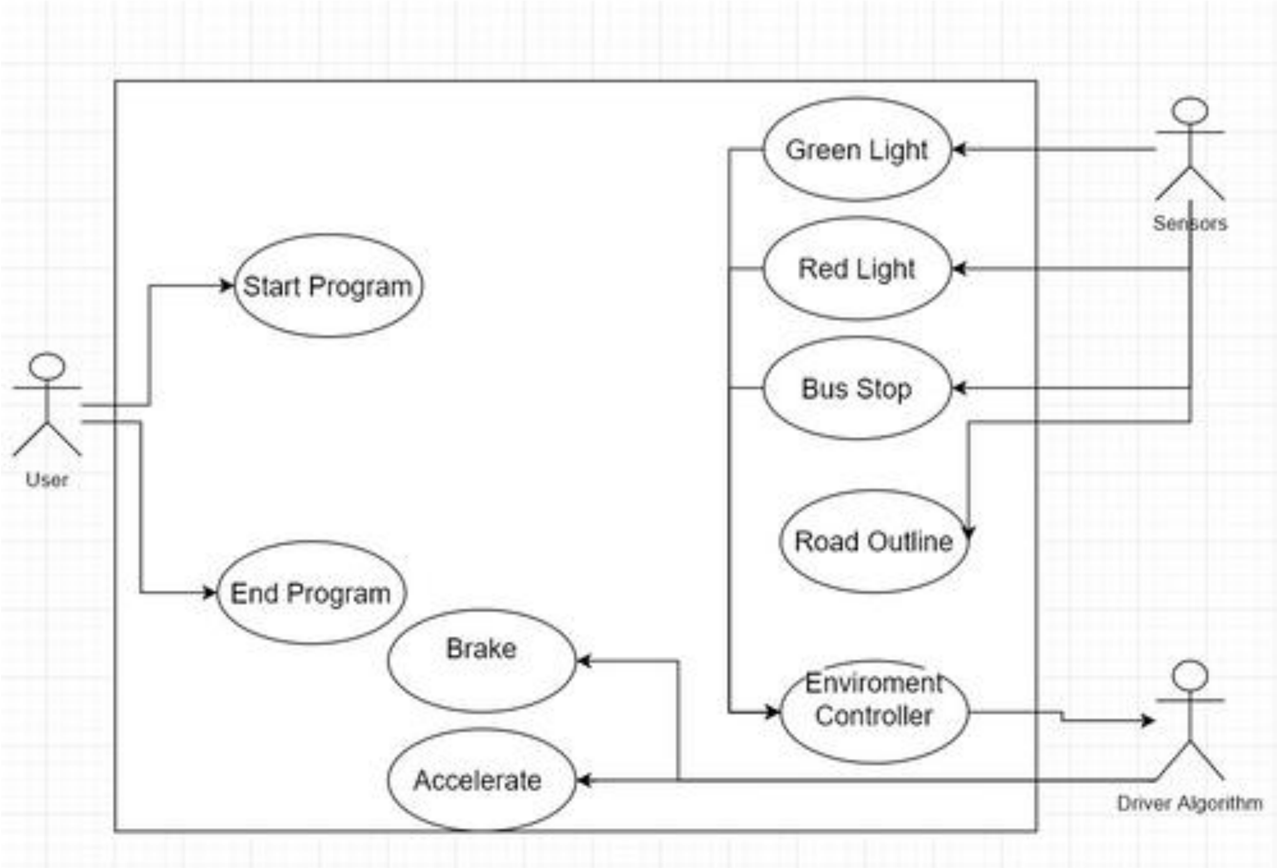
4.2.3 Architecture Design

4.2.3.1 Activity Diagram



.

4.2.3.2 Dataflow Diagram



4.2.3.3 Architecture Design of Automated Self Learning Bus Simulation

- **Case**: Start Program Overview User starts the simulation. Response The road is generated, bus starts its path, the sensors start and continues until simulation is ended.
- **Case**: End Program Overview User ends the simulation. Response Simulation is ended, total distance travelled and number of bus stops achieved is given to user.
- **Case**: Enviroment Controller Overview Sensors on the bus returns distance of objects that are in sensor range. Response Sensor data is given to the machine learning algorithm to decide the next response.
- **Case**: Brake Overview Slows the bus down when an obstacle or a red light appears. Response When sensors detect an obstacle or a red light the bus starts slowing down to increase its time to reach the object.
- **Case**: Accelerate Overview Increases the buses speed when sensors do not detect an obstacle. Response The bus starts to accelerate when the traffic light turns green or when an obstacle disapears from sensor range.

- **Case**: Stop Overview Slows the bus down to stop. Response If the bus gets close to obstacle or a red light despite slowing down it is stopped until the reason is cleared.
- **Case**: Red Light Overview Sensors detect a red light in front of the bus ands slows down. Response Slows the bus down when a red trafic light is seen in front of the bus and stops if the bus gets too close.
- **Case**: Green Light Overview Sensors detect a greenlight in front of te bus. Response Increases the buses speed when an obstacle is cleared or the red light turns green.
- **Case**: Bus Stop Overview Sensor detects a bus stop and changes route to it. Response Sensor detects a bus stop and changes route to stop on it.
- **Case**: Road Outline Overview Sensors measure the distance to road boundries. Response Sensors measure the distance to road boundries andsteers away if too close.

4.2.4 Simulation Environment

In our project, we will use Unity Engine to test our code so we will need an environment to observe the results.

Modelling Environment

The development environment of this project is Unity3D and some tools such as Blender and Maya. We will also use some assets in the Unity Asset Store to increase the speed of game development, such as cars, houses, obstacles, maps, level design and other things. For lighting, particle effects, sound effects and animation, we will use the Unity game engine. We will use low-poly design as they all use as few polygons as possible to define the shape of the object. In low-polygon design, you can get performance at the same time due to details and polygons. We need excellent performance to better train our bus agents.

4.2.5 References

[1]- Unity-Technologies. (n.d.). Unity-Technologies/ML-Agents: Unity Machine Learning Agents Toolkit. GitHub. Retrieved December 29, 2021, from https://github.com/Unity-Technologies/ml-agents

.

5. Conclusion

As a result, it can be said that the system was reviewed in the literature, the methods to be used were studied and reported, and the libraries to be used were decided. After a literature review, we prepared a software requirements specification document. Through SRS, information can be provided on many topics such as General purpose and functional requirements of the system SDD is a software design specification document that contains information about the system architecture. This report contains a lot of theory Information on Autonomous bus simulation system. In the future, a system overlay Theoretical information will be provided.