Patricia Annette C. Santos 15 January 2024 Word Count: 545
OOP_PCOM7E November 2023 Summative Assessment 1: System Design

—---

A Design Proposal of Software to Support Operation of a Driverless Car

Introduction

The assignment is a design proposal of software using object-oriented Python programming principles applied to three driverless car operations that an actor has the ability to engage with through the front-end. The three operations selected are: *Vehicle to Infrastructure Communication, Environment Perception,* and *Car Control.* The design and operation of the Autonomous Vehicle is presented using Unified Modeling Language (UML).

Keywords: Autonomous Vehicle (AV), Driverless Car, Vehicle to Infrastructure (V2I) Communication, Environment Perception, Car Control, Mapping, Light Detection and Ranging (LiDAR)

UML Models for the Proposed Design

Class diagram

The super class is the ‘Driverless Car’ initialised by user log-in. The subclasses are ‘Car Control’, ‘Environment Perception’, and ‘V2I’. According to Serban et al. (2020), the majority of data in AVs should be relayed at the same time they are acquired. For the proposal, this is relayed through the V2I during Environment Perception. Both functions have to be working before the driverless car can control itself. Control systems are depended on by AVs to help them instantaneously select the suitable steps during driving based on gathered data (Khare & Jain, 2023).

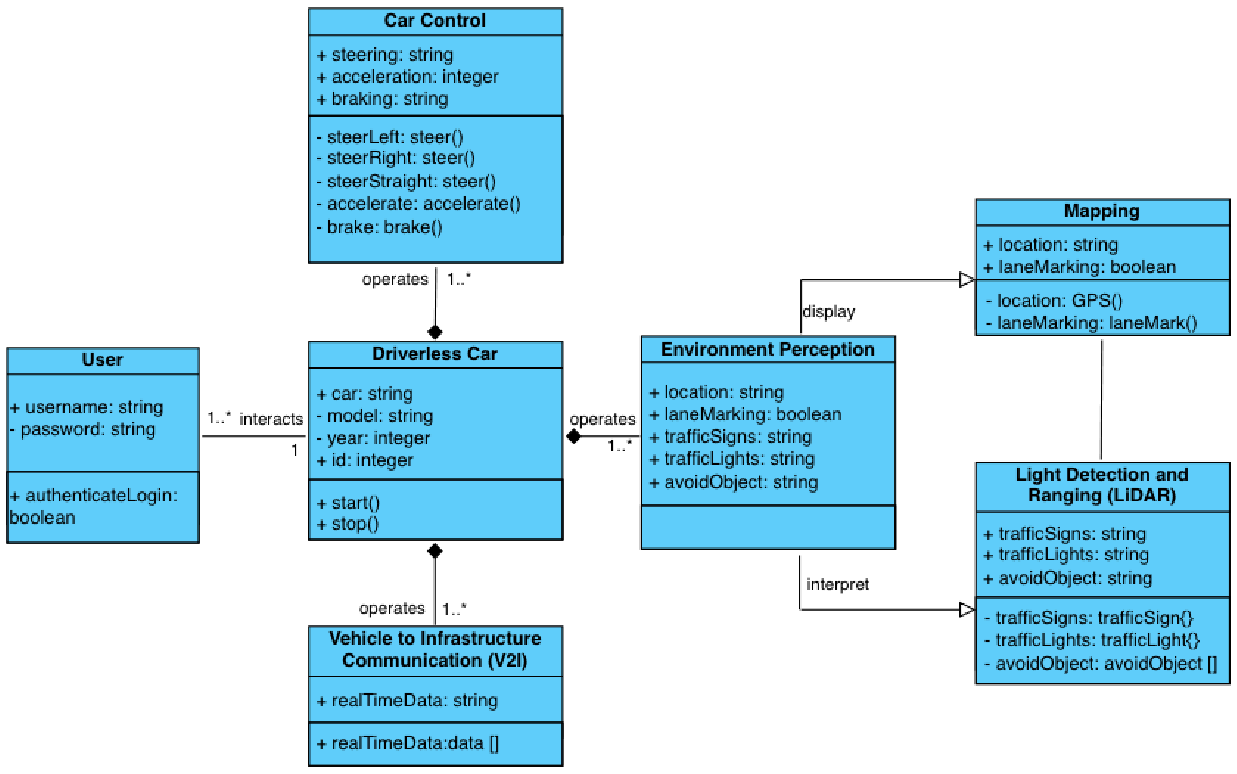


Figure 1: Class Diagram

Use case diagram

The user has to login first in order to start the AV. Once authenticated, the driverless car can start and stop automatically. The function that steers and operates driverless cars autonomously is ‘Car Control’. Some of the Car Control tasks listed by Reddy (2019) include speed and direction. In addition, appropriate steering and acceleration are ways of safely controlling AV movement (Bachute & Subhedar, 2021).

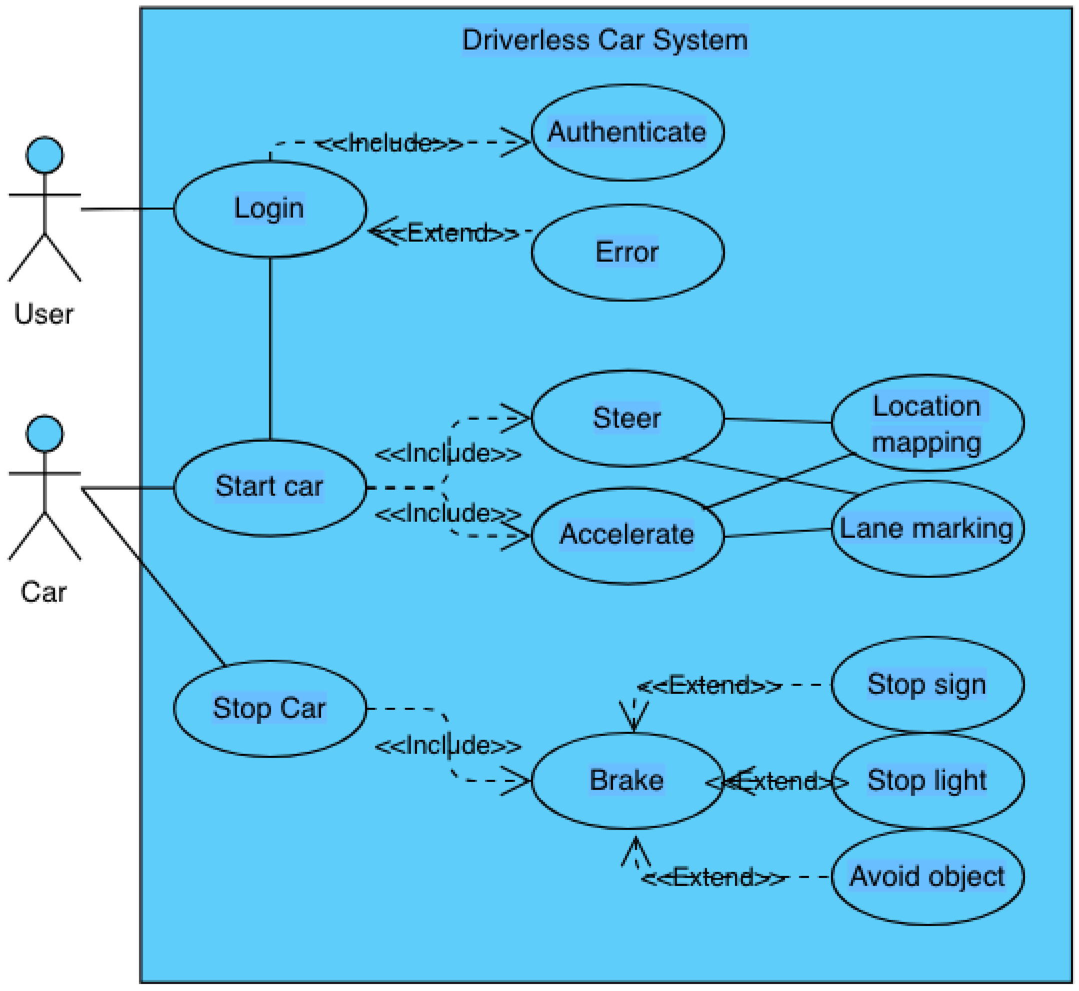


Figure 2: Use Case Diagram

Activity diagram

Khare and Jain (2023) explain that concurrent information shared collaboratively through V2I can prevent road traffic accidents. They also add that maps relay important information like location, as well as road marks and signs, among others.

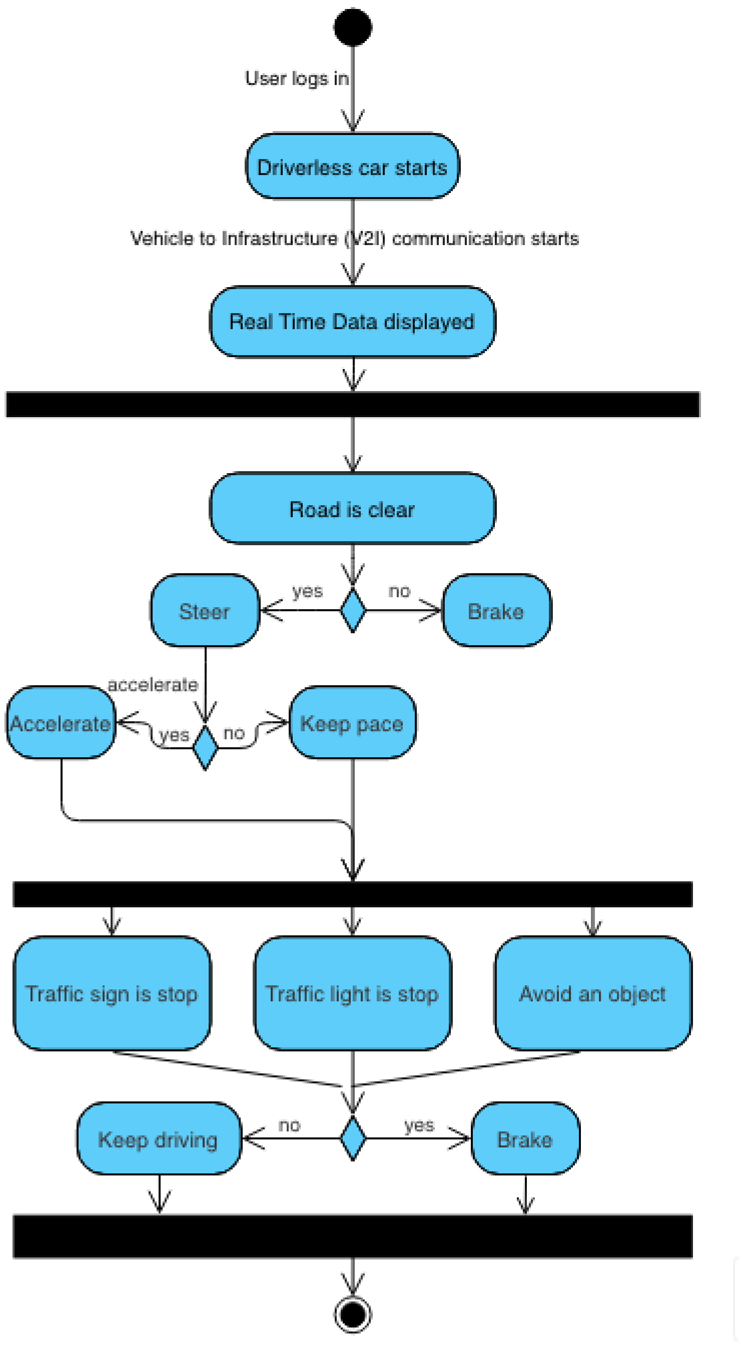


Figure 3: Activity Diagram

Sequence diagram

Reddy (2019) explains key technologies in automated cars, and describes ‘Environment Perception’ as one that senses elements around it using various navigation tools. Zhou and Sun (2019) deep dives into the use of the light detection and ranging technology known as LiDAR, that perceives even faraway objects to make out the environment. Data gathered through Mapping and LiDAR helps AVs decide whether to steer, accelerate, or brake.

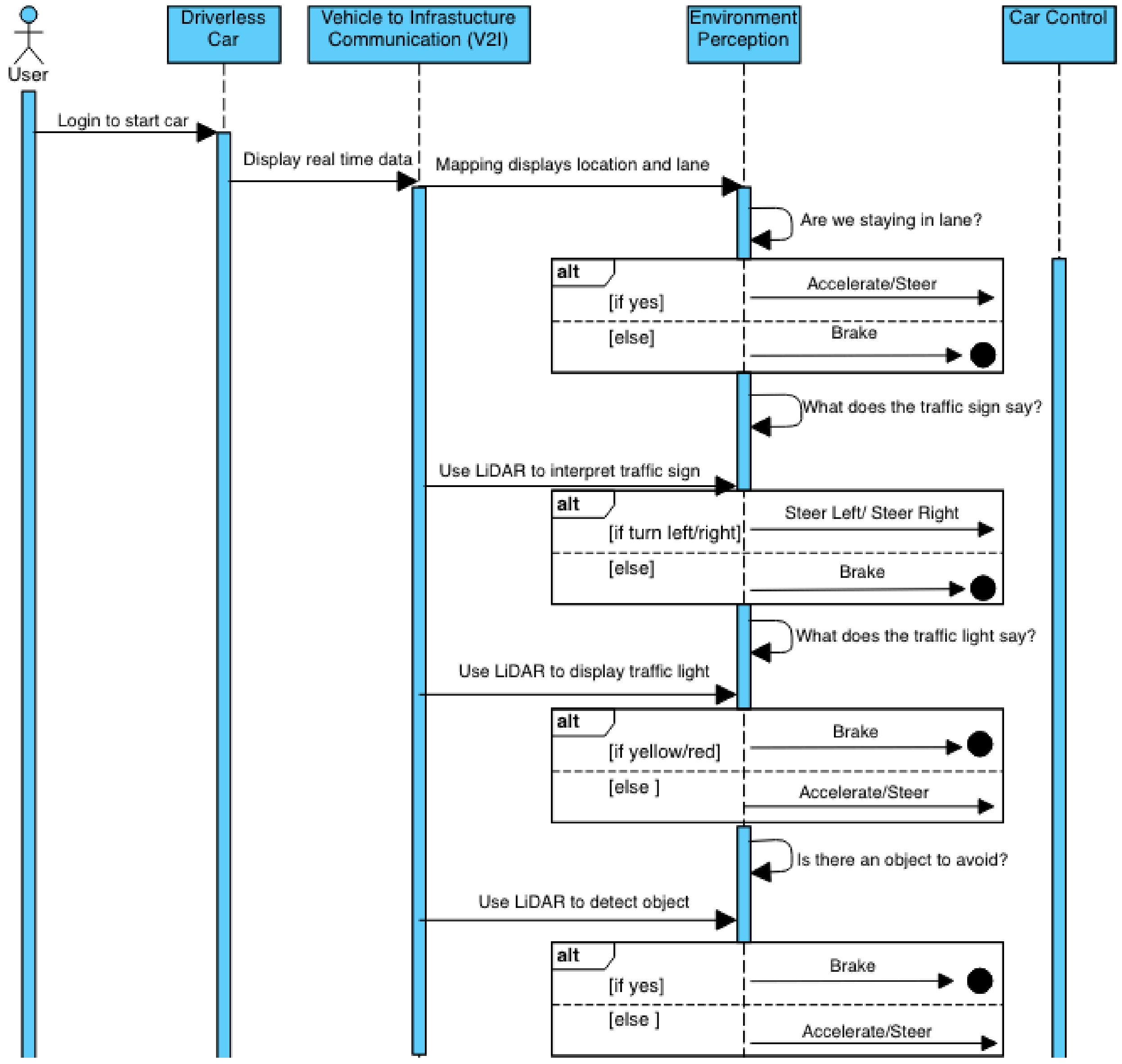


Figure 4: Sequence Diagram

State Transition diagram

'V2I Communication' takes advantage of wireless network and technology to process gathered information from traffic lights, signs, and lanes (Kanthavel et al., 2021).

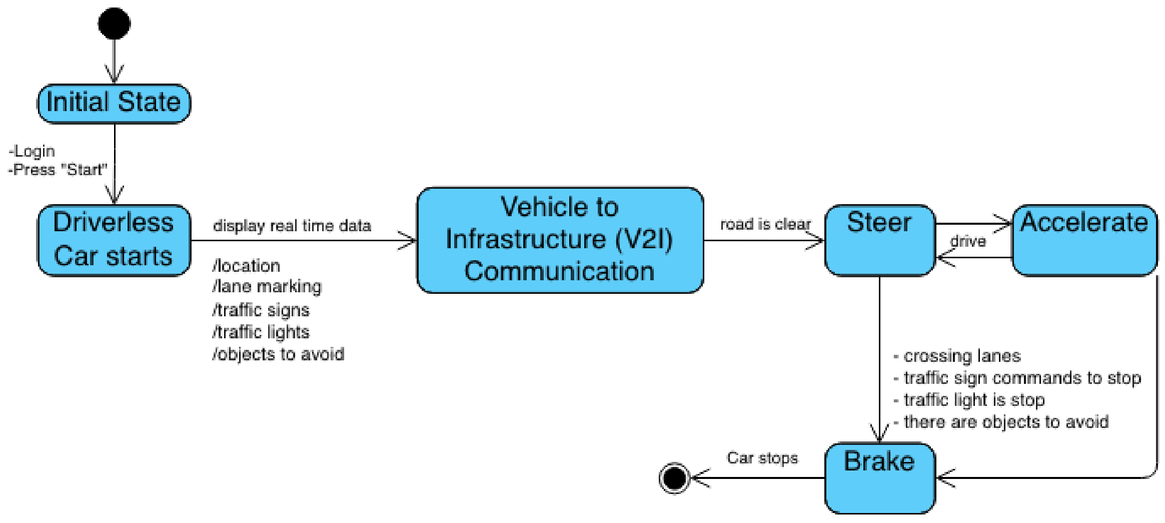


Figure 5: State Transition Diagram

Retaining Data Within Data Structures

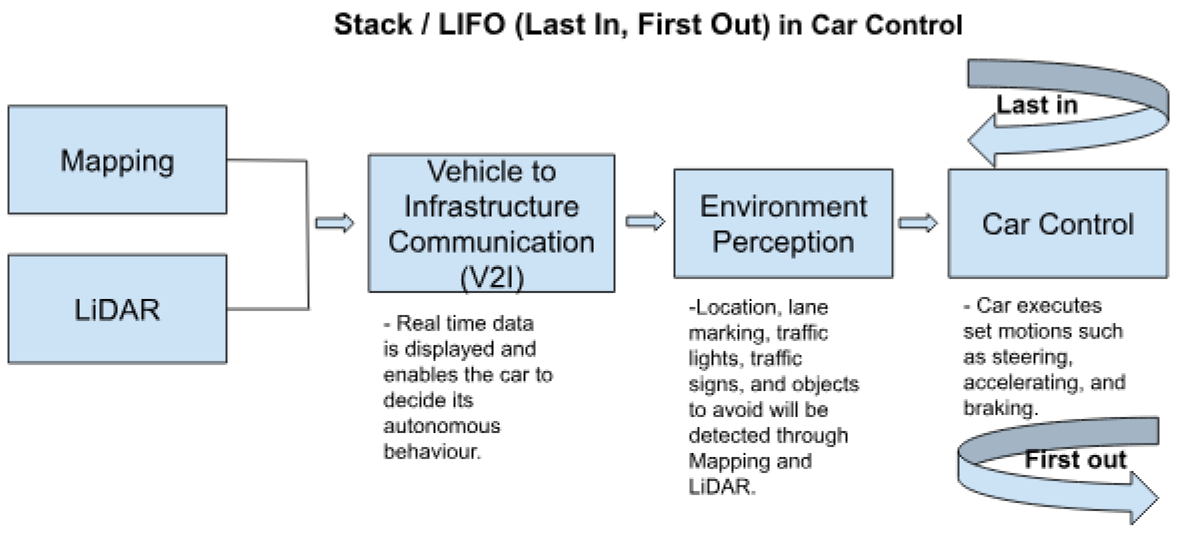

Stack / LIFO (Last In, First Out) in Car Control
Mapping
LiDAR
Vehicle to Infrastructure Communication (V2I)
- Real time data is displayed and enables the car to decide its autonomous behaviour.
Environment Perception
-Location, lane marking, traffic lights, traffic signs, and objects to avoid will be detected through Mapping and LiDAR.
Car Control
- Car executes set motions such as steering, accelerating, and braking.
Last in
First out

Figure 6: Stack

This data structure is based on the AV software stack order presented by Silver (2018) as a basis for the Last In, First Out (LIFO) data acquisition.

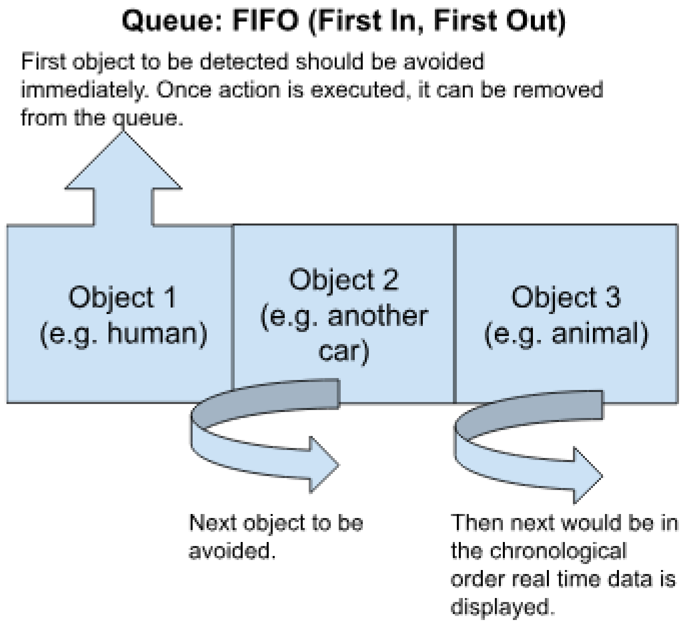
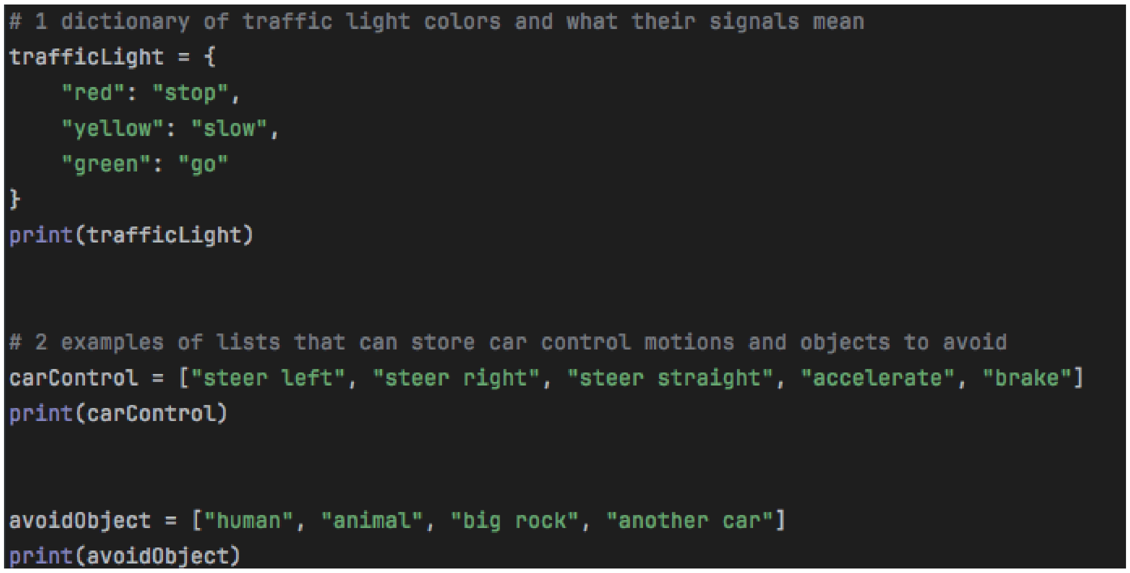

Queue: FIFO (First In, First Out)
First object to be detected should be avoided immediately. Once action is executed, it can be removed from the queue.
Object 1 (e.g. human)
Object 2 (e.g. another car)
Object 3 (e.g. animal)
Next object to be avoided.
Then next would be in the chronological order real time data is displayed.

Figure 7: Queue

Real time reports of objects to avoid are stored in a queue as they provide up-to-date data to the AV. Once acted upon, they can then be removed from the queue using the append method also known as First In, First Out (FIFO).



```
# 1 dictionary of traffic light colors and what their signals mean
trafficLight = {
    "red": "stop",
    "yellow": "slow",
    "green": "go"
}
print(trafficLight)


# 2 examples of lists that can store car control motions and objects to avoid
carControl = ["steer left", "steer right", "steer straight", "accelerate", "brake"]
print(carControl)


avoidObject = ["human", "animal", "big rock", "another car"]
print(avoidObject)
```

Figure 8: Dictionary and List

Dictionary is utilised for traffic signs and lights. One traffic light or sign would entail just one specific action, making paired keys and values effective in storing such data. A list of pre-selected objects to avoid, based on machine learning, and the predetermined car control motions, are stored in a list as lists keep element records.

References:

Bachute, M.R. & Subhedar, J.M. (2021) Autonomous Driving Architectures: Insights of Machine Learning and Deep Learning Algorithms. *Machine Learning with Applications* 6(100164): 1-25. DOI: https://doi.org/10.1016/j.mlwa.2021.100164.

Kanthavel, D., Sangeetha, S.K.B. & Keerthana, K.P. (2021) An empirical study of vehicle to infrastructure communications- An intense learning of smart infrastructure for safety and mobility. *International Journal of Intelligent Networks.* 2: 77-82. DOI: https://doi.org/10.1016/j.ijin.2021.06.003.

Khare, V. & Jain, A. (2023) Predict the performance of driverless car through the cognitive data analysis and reliability analysis based approach. *e-Prime-Advances in Electrical Engineering, Electronics and Energy.* 6(100344): 1-10. DOI: https://doi.org/10.1016/j.prime.2023.100344.

Reddy, P.P. (2019) Driverless Car: Software Modelling and Design Using Python and Tensorflow. Available from: https://easychair.org/publications/preprint_open/k7wj [Accessed 07 January 2024].

Serban, A., Poll, E. & Visser, J. (2020). A Standard Driven Software Architecture for Fully Autonomous Vehicles. *Journal of Automotive Software Engineering.* 1(1): 20-33. DOI: https://doi.org/10.2991/jase.d.200212.001.

Silver, D. (2018). How Control Works for Self-Driving Cars. Available from: https://www.linkedin.com/pulse/how-control-works-self-driving-cars-david-silver/ [Accessed 15 January 2024].

Zhou, Z. Q. & Sun, L. (2019) Metamorphic Testing of Driverless Cars. *Communications of the ACM* 62(3): 61-67. DOI: https://doi.org/10.1145/3241979.

Bibliography:

Everything Computer Science, Inc. (N.D.) Stacks and Queues. Available from: https://everythingcomputerscience.com/discrete_mathematics/Stacks_and_Queues.html [Accessed 15 January 2024].

Programiz (N.D.) Python List (With Examples). Available from: https://www.programiz.com/python-programming/list [Accessed 15 January 2024].

Programiz (N.D.) Python Dictionary (With Examples). Available from: https://www.programiz.com/python-programming/dictionary [Accessed 15 January 2024].

Python (N.D.) 5.Data Structures – Python 3.12.1 documentation. Available from: https://docs.python.org/3/tutorial/datastructures.html [Accessed 11 January 2024].

Rumbaugh, J., Jacobson, I. & Booch, G. (2005) *The Unified Modeling Language Reference Manual, Second Edition.* Massachusetts: Pearson Education. Available from: https://learning.oreilly.com/library/view/unified-modeling-language/0321245628/ [Accessed 10 January 2024].