Faculty of Science and Technology

BSc (Hons) Games Programming

May 2019

Using Artificial Intelligence within Cars to Understand the Awareness of Road Systems and Improve Traffic Errors

By

Alfie Scully

**Abstract**

**Table of Contents**

[Chapter 1: Introduction 4](#_Toc1043356)

[**1.0** **Concept and Justification of Project** 4](#_Toc1043357)

[**1.1 Aims and Objectives** 4](#_Toc1043358)

[**1.1.1 Aims** 4](#_Toc1043359)

[**1.1.2 Objectives** 4](#_Toc1043360)

[Chapter 2: Literature Review 5](#_Toc1043361)

[**2.0** **Introduction** 5](#_Toc1043362)

[**2.1 Road Management System** 5](#_Toc1043363)

[**2.2 Hierarchy of Decision-making Processes** 5](#_Toc1043364)

[**2.2.1 Route Planning** 6](#_Toc1043365)

[**2.2.2 Behavioural Layer** 7](#_Toc1043366)

[**2.2.3 Motion Planning** 8](#_Toc1043367)

[**2.2.4 Vehicle Control** 8](#_Toc1043368)

[**2.3 AI Agents** 9](#_Toc1043369)

[**Summary** 9](#_Toc1043370)

[References 10](#_Toc1043371)

# **Chapter 1: Introduction**

## **Concept and Justification of Project**

**….**

## **1.1 Aims and Objectives**

### **1.1.1 Aims**

The aim of this project is to develop a 3D simulation of a controllable management system for self-driving cars to deal with roads and traffic problems, using Artificial intelligence algorithms that support the cars to function with handling the driving abilities.

This will focus on dealing with road management systems such as traffic lights and speed limits, and awareness of another vehicle presence near each other. The aim is to get cars to gather information in real-time and respond to the systems for traffic lights and speed limits. The project will then proceed onto dealing with traffics issues that are common on the road such as tailgating, gridlocks and intersections.

### **1.1.2 Objectives**

To achieve the project aim, the project will need to focus on the following objectives:

Primary objectives:

* Research and analyse algorithms that may be used into the simulation and decide which algorithms are best for the project.
* Create the AI algorithm that navigates the car around the road environment.
* Create the algorithm solution to the problem of cars having to account for the actual road systems: e.g. dealing with the traffic lights and speed limits.
* Implement variations of driver behaviours to be included within the vehicles’ driving solution:
* Safe drivers
* Road rage
* Speeding
* Harsh braking
* Implement the ability of the cars to assess and react to other cars on the road network to fix traffic problems: e.g. tailgating, gridlocks and red-light runners.

Secondary objectives (extensions if possible):

* Implement a system of night and day cycles which the artificial intelligence cars reacts to and affect what the cars will do.
* Add extra vehicle types, and their specific features. For example, Busses having a cycle of bus stops to navigate to.

# **Chapter 2: Literature Review**

**Background Theory**

**……..**

## **Introduction**

The following literature review will identify information that will be used to discuss the topics and techniques that relates to this project, which the information researched will have relevancy to support the implementation methods and help carry out the progress of this project. Then an evaluation will be performed to decide on which methods and techniques are most appropriate to help proceed with what has been defined within the full proposal.

## **2.1 Road Management System**

## **2.2 Hierarchy of Decision-making Processes**

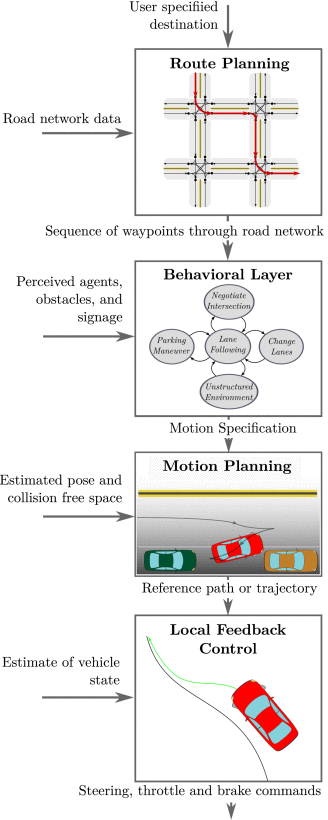
The Hierarchy of Decision-Making (HDM) is a series of layers used within driverless cars that helps the car drive on the road using a decision-making system. The system is hierarchically sectioned into four components with the first layer being Route Planning using the road network data. Then followed by Behavioural Layer, which is used to help the car understand the rules of the road. After that Motion Planning is utilized and what this does is help the car understand what is around the environment during its path. Finally, the last layer being Local Feedback Control which will correct the errors that execute from Motion planning. (Paden et al. 2016)

Figure 1.1 Illustration of the hierarchy of decision-making processes (Paden et al. 2016)

### **2.2.1 Route Planning**

In HDM the highest level is route planning, and this provide the ability to the car to be able to have a decision-making system that selects a route on a road network from the current point to the destination point. The road network will be represented as a graph which was mentioned above [WRITE PARAGRAPH ABOVE], for the route to be formulated pathfinding algorithms such as Dijkstra and A\* will be heavily used to help find a minimum cost path on a road network graph. (Paden et al. 2016)

The Dijkstra algorithm published in 1959 is best known for dealing with weighted graphs and finding the shortest path from current node to the destination node using a graph. (Abiy et al. 2019) The algorithm will look for which path has the lowest weight on each edge and are all non-negative values. (<http://cs.indstate.edu/hgopireddy/algor.pdf>)

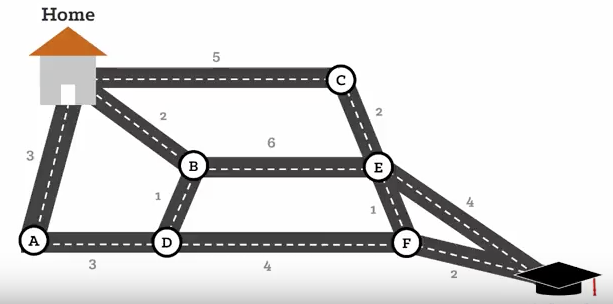
The shortest path that is found in this example is, Home, B, D, F, School.

Figure 1.2 Dijkstra's Shortest Path Algorithm (Abiy et al. 2019)

A\* was invented in 1964 by Nils Nilsson(<https://ieeexplore.ieee.org/document/4082128>), it is another algorithm used in pathfinding and graph traversal. A\* introduces a heuristic approach in graph searching algorithms, it can plan ahead in each step to create an optimal path. (<https://brilliant.org/wiki/a-star-search/>)

An example of the benefits of using A\* is shown below:

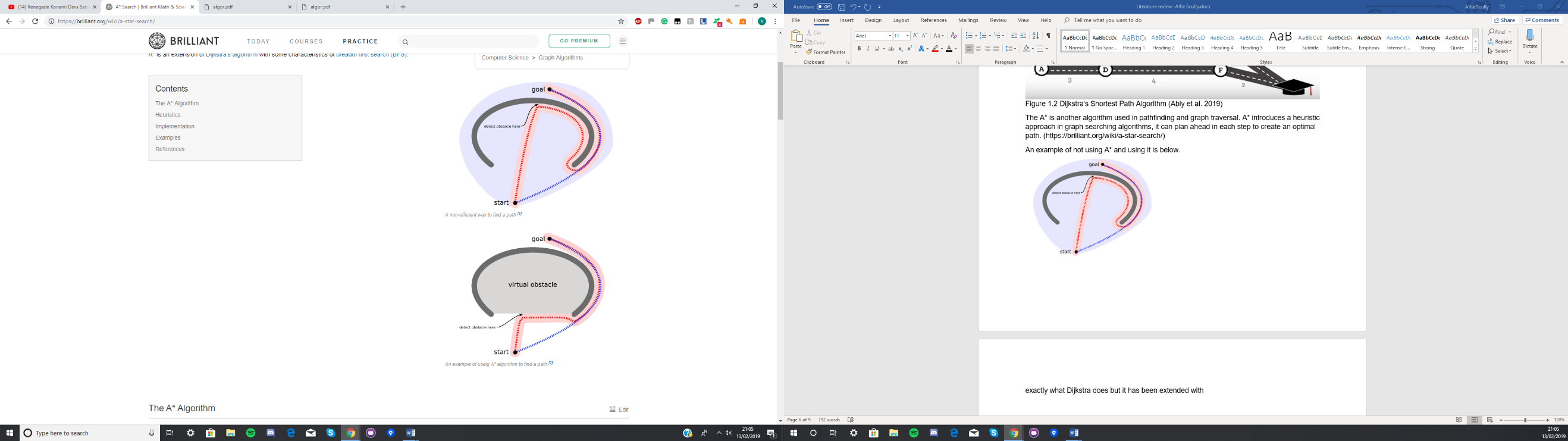
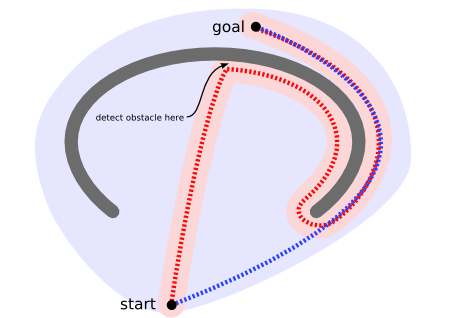


Figure 1.3 A\* Search (<https://brilliant.org/wiki/a-star-search/>)

Figure 1.3.1 Unoptimized Search (……………) Figure 1.3.2 A\* Search (……………)

As seen from figure 1.3, A\* is able to create a more optimal and quicker path, Also A\* is an extension of how Dijkstra operates however it has the characteristics of breadth-first search. (<http://cs.indstate.edu/hgopireddy/algor.pdf>)

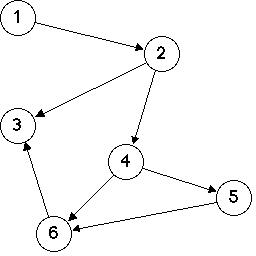
Although Dijkstra is optimal and finds the shortest path, the issues that occur with using this algorithm is that it only see’s the neighbouring nodes and since it does not keep a backtrack of previous nodes, it could potentially lead into an infinite loop and waste necessary resources. Also, it cannot handle negative edges and this makes the graph acyclic, so the shortest path will not be obtained. (<http://cs.indstate.edu/hgopireddy/algor.pdf>)

Figure 1.4 acyclic graph (https://www.cs.hmc.edu/~keller/courses/cs60/s98/examples/acyclic/)

The main issue with A\* algorithms is that it is a best-first search, and this means it has memory requirements. So, this means it saves a list of all previous nodes it would have passed during the pathfinding process and so the algorithm is space limited. (<http://cs.indstate.edu/hgopireddy/algor.pdf>)

Overall both Dijkstra’s and A\* algorithm are great for route planning, however the most optimal algorithm out of the two is the A\* Algorithm. This is because it doesn’t only find shortest path, but it finds the quickest path and the issue with using Dijkstra is that it would not be as fast or optimal compare to A\* if route planning was in a dense environment such as a city road network (<https://www.youtube.com/watch?v=ySN5Wnu88nE&t=396s>), so in this project A\* will be implemented.

### **2.2.2** **Behavioural Layer**

The second highest layer in HDM after a route is planned, is the behavioural layer. What the layer will perform for the car is it will interact with other participants, e.g. cars and the rules set on the road. Furthermore, the cars will select a certain driving behaviour based upon the perceived behaviour of other cars, conditions of the roads, and road infrastructure. An example of this would be when the car comes to a traffic light and it is red, the layer will command the car to stop at the light, observe other car behaviours and then let the vehicle proceed on driving. (Paden et al. 2016)

So, for this layer the project will be looking into the use of behaviour trees and finite state machines.

Behaviour Trees

* What are they
* Show example
* Good
* Bad
* Mention the possibility of using finite state machine instead
* But overall decide to use behaviour trees

A Survey of Behaviour Trees and their Applications for Game AI

<https://www.google.com/search?q=issue+iwth+beahiour+trees&oq=issue+iwth+beahiour+trees&aqs=chrome..69i57.6366j0j7&sourceid=chrome&ie=UTF-8>

Behaviour Trees for decision-making in autonomous driving <http://www.diva-portal.org/smash/get/diva2:907048/FULLTEXT01.pdf>

### **2.2.3 Motion Planning**

Will use motion radars to help with this process

<https://ieeexplore.ieee.org/abstract/document/7490340>

### **2.2.4 Vehicle Control**

Correct the errors from executions of motion planning

--- link helpful

<http://www.diva-portal.org/smash/get/diva2:907048/FULLTEXT01.pdf>

**2.1 Road Management System**

Road Management System:

This part is going to mention the management system of the road, e.g. rules.

UK road and traffic sign symbols

<https://www.highwaycodeuk.co.uk/signs-and-signals.html>

UK road system

<https://www.gov.uk/topic/transport/traffic-road-management>

Using Game Engines for Designing Traffic Control Educational Games <https://ieeexplore.ieee.org/document/7313131?fbclid=IwAR3-OK1hh-5qHEtpcn6DvVJVHLAgYKZ_GGazgrtebNAr1-V5huEpzLjNl_c>

A dynamic and automatic traffic light control expert system for solving the road congestion problem

<https://www.sciencedirect.com/science/article/pii/S0957417407001303>

Graph Traversal

Represent the road network as a graph structure

https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/

## **2.3 AI Agents**

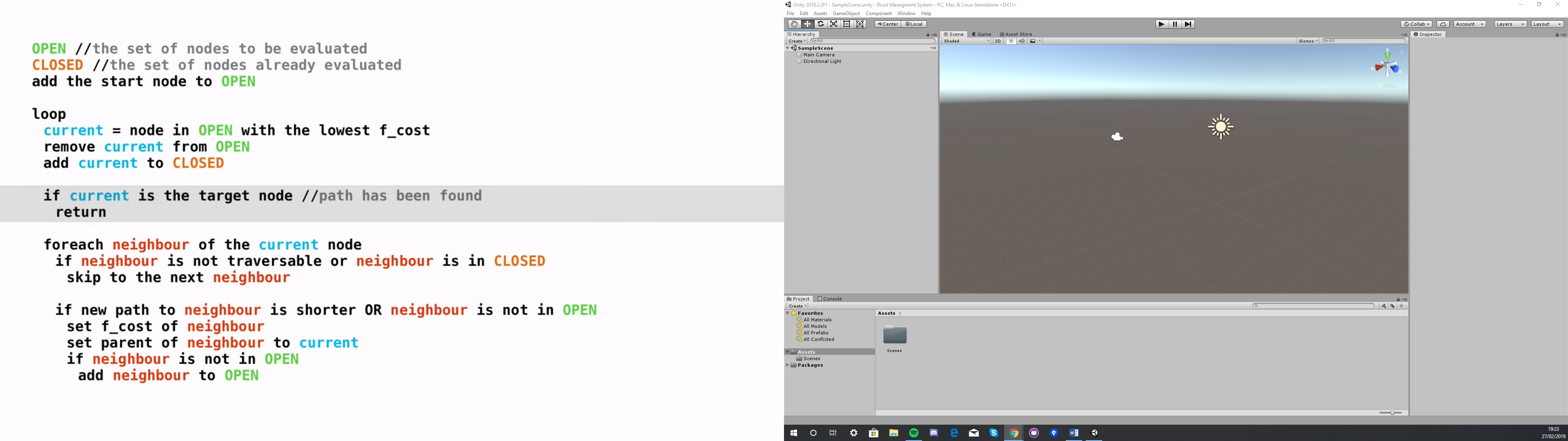
Intelligent agents: theory and practice

<https://www.cambridge.org/core/journals/knowledge-engineering-review/article/intelligent-agents-theory-and-practice/CF2A6AAEEA1DBD486EF019F6217F1597>

A Review of the Applications of Agent Technology in Traffic and Transportation Systems

<https://ieeexplore.ieee.org/abstract/document/5462881>

## **Summary**



<https://www.codeproject.com/Articles/1221034/Pathfinding-Algorithms-in-Csharp>

<https://www.researchgate.net/post/How_to_get_a_city_map_into_a_graph_format_of_nodes_and_edges>

<https://study.com/academy/lesson/graphs-in-discrete-math-definition-types-uses.html>

<https://simpledevcode.wordpress.com/2015/12/22/graphs-and-dijkstras-algorithm-c/>

<https://simpledevcode.wordpress.com/2015/08/10/priority-queue-tutorial-c-c-java/>

Look into undirected graph structure or visibility graph.

# **References**

(Paden et al. 2016)

Paden, B., Cap, M., Yong, S., Yershov, D. and Frazzzoli, E., 2016. A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles - IEEE Journals & Magazine [online]. Ieeexplore.ieee.org. Available from: https://ieeexplore.ieee.org/abstract/document/7490340 [Accessed 1 Nov 2018].

(Abiy et al. 2019)

Abiy, T., Pang, H. and Williams, C. (2019). Dijkstra's Shortest Path Algorithm | Brilliant Math & Science Wiki. [online] Brilliant.org. Available at: https://brilliant.org/wiki/dijkstras-short-path-finder/ [Accessed 8 Feb. 2019].