

Computer Games Development SE607

Software Functional Specification

Year IV

|  |  |
| --- | --- |
| Donal Howe  C00249662 | [Student Name]  [Student Number] |
| [Date of Submission] | |

[Declaration form to be attached]

Contents

[Acknowledgements 2](#_Toc54713736)

[Functional Specification 2](#_Toc54713737)

[References 2](#_Toc54713738)

# Acknowledgements

We would like to thank the following people who assisted in completing this project including;

John Doe of ACME who kindly agreed to …

I would also like to thank #### Solutions for use of ….

# Introduction

The objective of this project was to put together a comprehensive comparison of guided and non-guided based pathfinding algorithms to the incremental dynamic pathfinding algorithm known as Dstar Lite under a game’s development context. So one can decide based of the information shown in this document whether to or not implement Dstar lite into their project or perhaps to implement another algorithm such as life long planning A star or, A star itself.

Description of D star Lite

Dstar Lite works as a dynamic A star where it can make changes to the path along the graph without having to rerun the process of calculating the path. Where Astar has to calculate the heuristic (cost of the node form the destination + the cost of the node from the beginning node) for each node upon running the algorithm to find the shortest path D star does not. It works by only investigating nodes which have been affected by a non-traversable which has been placed on the path. This in turn makes rerunning the algorithm potentially cheaper than having to recalculate the entire path.

Description of A star

Astar is a heuristic algorithm being that it knows the end and start point. It then tries to find the shortest path to the end point, however it will rerun itself if an obstacle gets in the way. Astar can find the shortest path through a priority queue which will compare the values of each node using both their Hcost( distance from the node) and Gcost( distance from the start node). This is how it knows to look at certain nodes first.

# Functional Specification

The software will in essence function as a visual pathfinding application. So the user will run the application and see a basic grid they can then adjust the size of the grid to three specified sizes “Small” being a 10x10 grid , “Medium” being a 50x50 grid and “Large” being a 100x100 size grid. They can then choose from a variety of pathfinding algorithms them being Astar, Dstar Lite, Djkstras algorithm, life long planning Astar , jump point search and the only no heuristic pathfinding algorithm depth first search. The user can also place down obstacles during process of the algorithms search and before the algorithm has been ran if they perhaps are looking for a specific path, onto the grid which will have the pathfinding algorithms react to them and find a corresponding path.

Chart

Description automatically generated The user can select grid size and the algorithm they wish to use.

A picture containing shoji, building

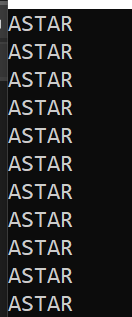
Description automatically generated The small grid with 100 cells and row of columns of 10 each

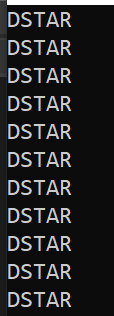
Background pattern

Description automatically generated The large grid with 2500 cells and row of columns of 50 each

Background pattern

Description automatically generated The large grid with 10,000 cells and row of columns of 100 each

 The Astar Search algorithm being ran and displayed in the console there will be a visual representation in the application.

 The Dstar Lite Search algorithm being ran and displayed in the console there will be a visual representation in the application.

A picture containing text

Description automatically generated The Jump Point Search algorithm being ran and displayed in the console there will be a visual representation in the application.

Text

Description automatically generated The Lifelong Planning Astar (LPA\*) algorithm being ran and displayed in the console there will be a visual representation in the application.

Text, chat or text message

Description automatically generated The Depth First Search algorithm being ran and displayed in the console there will be a visual representation in the application.

Text

Description automatically generated The Djkstras Search algorithm being ran and displayed in the console there will be a visual representation in the application.

A screenshot of a computer screen

Description automatically generated with medium confidence My Astar function compute shortest path function

Text

Description automatically generated The Main Dstar lite function which controls the wall handling and progression of the robot position on the path

Text

Description automatically generated The compute shortest path function which controls the handling of the different types of inconsistent nodes being over consistent and under consistent.

Text

Description automatically generated The initialise Dstar function initialises all of the nodes in the grid to suit for the algorithm to work and pushes the goal node into the priority queue with the correct RHS cost and Gcost

Text

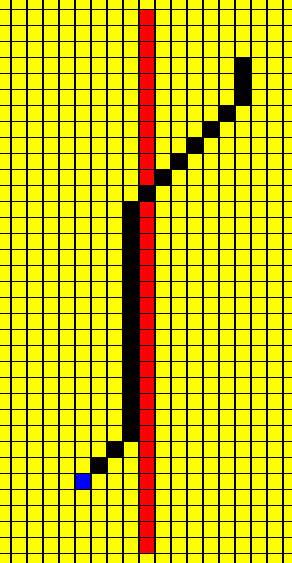
Description automatically generated The Dstar Lite Update Vertex function which updates the key costs of nodes and potentially pushes them into the queue for further investigation

Text

Description automatically generated The calculation of the D star key

A picture containing shoji, building, silhouette

Description automatically generated an example of basic walls. Red Nodes are the walls placed on the grid. The grid size in question is the small grid of size 100 nodes.

 path returned using Astar

Design and describe how the application will be used: e.g. Navigation sequence in Web based project.

The user will be met with a grid will ask which algorithm they want to use. They can then place obstacles as they wish and as such run the pathfinder. They will then get a string pop up on the screen which will suggest the better algorithm for the scenario either Astar or Dstar.

User interfaces should be specified loosely here (possibly screen shots of prototype user interfaces - you will not be held to using these UIs they are just to assist in explanation of your project functionality).

Include any element that you would feel conveys how you perceive your project will function.

# References

# References

<https://core.ac.uk/download/pdf/235050716.pdf> - Path Planning Algorithm using D\* Heuristic Method Based on PSO in Dynamic Environment Firas A. Raheema \*, Umniah I. Hameedb

<https://medium.com/@nicholas.w.swift/easy-a-star-pathfinding-7e6689c7f7b2> -Nicholas Swift Feb 27 2017

<http://www.cs.cmu.edu/~ggordon/likhachev-etal.anytime-dstar.pdf> - Maxim Likhachev† , Dave Ferguson† , Geoff Gordon† , Anthony Stentz† , and Sebastian Thrun‡

<https://www.ri.cmu.edu/pub_files/pub3/stentz_anthony__tony__1994_2/stentz_anthony__tony__1994_2.pdf> -Anthony Stentz