

Lesson 3:
A* Search

SEARCH

RESOURCES

CONCEPTS

1. Intro

2. Motion Planning

3. Maze

4. Maze 2

5. Coding the Shortest Path Algorit...

6. A* Search

7. Lesson Code Structure

8. CODE: Starting A* Search

9. CODE: Writing the A* Heuristic

10. Pass by Reference in C++

11. CODE: Adding Nodes to the Ope...

12. CODE: Initialize the Open Vector

13. CODE: Create a Comparison Fun...

14. CODE: Write a While Loop for the...

15. CODE: Check for Valid Neighbors

16. Constants

17. CODE: Expand the A* Search to ...

18. Arrays

19. CODE: Adding a Start and End to ...

20. Congratulations!!

21. How to Become More Proficient ...

Coding the Shortest Path Algorithm

SEND FEEDBACK

Coding the Shortest Path Algorithm

In the previous videos, Sebastian talked about the motion planning problem, and you worked through a couple of exercises using a grid-based version of this problem. In this next video, Sebastian will describe how a general path search works in a grid.



The diagram illustrates a grid-based search problem. It features a 5x5 grid with several cells shaded to represent obstacles. A start cell 'S' is located at the top-left corner (row 1, column 1), and a goal cell 'G' is at the bottom-right corner (row 5, column 5). To the right of the grid, four arrows point in the cardinal directions: UP, DOWN, LEFT, and RIGHT, indicating the possible movement directions for the search algorithm.

https://video.udacity-data.com/topher/2019/january/5c3e92e8_grid/grid_720p.mp4

NEXT