

Lesson 3:  
A\* Search

SEARCH

RESOURCES

CONCEPTS

1. Intro

2. Motion Planning

3. Maze

4. Maze 2

5. Coding the Shortest Path Algorithm

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 Open List

12. CODE: Initialize the Open Vector

13. CODE: Create a Comparison Function

14. CODE: Write a While Loop for the Search

15. CODE: Check for Valid Neighbors

16. Constants

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

18. Arrays

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

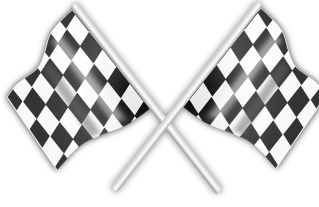
20. Congratulations!!

21. How to Become More Proficient with C++

CODE: Adding a Start and End to the Board

SEND FEEDBACK

Adding a Start and End to the Board



Excellent work! Your project is essentially complete, and the A\* search algorithm is fully functional. To wrap things up, there is one modification that can be made to the project to make the printout slightly clearer. At this point, your program should print the following:

```
  0 0 0 0 0
  0 0 0 0 0
  0 0 0 0 0
  0 0 0 0 0
  0 0 0 0 0
```

This is fantastic, but it isn't clear where the beginning and end of the path are. In this exercise, you will add a `0` for the beginning of the path, and a `1` for the end.

To Complete This Exercise:

- Add a `kStart` and `kFinish` to the `State` enum.
- Set the grid cell to `kStart` for the initial coordinates and `kFinish` for the goal coordinates. This will happen in the `Search` function.
- In `CellString`, add cases to return `"0 "` for `kStart` and `"1 "` for `kFinish`.

< +

/> home > workspace

1.board

main.cpp

solution.cpp

test.cpp

main.cpp

```
1 • #include <algorithm> // for sort
2 #include <iostream>
3 #include <iostream>
4 #include <sstream>
5 #include <string>
6 #include <vector>
7 using std::cout;
8 using std::ifstream;
9 using std::istringstream;
10 using std::sort;
11 using std::string;
12 using std::vector;
13 ...
14 ...
15 ...
```

+ BASH

root@4c1e850b9655: /home/workspace#

↑ Menu

↗ Expand

NEXT