# Pathfinding Implementations (DFS, BFS, A*)

Cole Kendall

## Context

Pathfinding is a common problem solved using computer science. It is used in video games, web crawlers, and mapping tools. I have done an implementation in the past; however, I was not happy with the quality of the end product. I would rather redo the project in a manner that is more presentable.

Since I need to program an "environment" for my pathfinder to traverse through, I plan to exercise the use of unit tests to validate the stability of my code.

## Objective

To design, implement, present, and host the code needed to perform a DFS, BFS, and A* implementation.

## Execution Notes

- At first, I over engineered the objects that would make up the environment. I reduced the amount of redundant code and simplified the overall structure of the code.
- I made sure to try and handle all possible inputs given to each method and write tests to ensure they are handled appropriately.
- I wrote out unit tests that reflected the desired behavior of each function. In this way, I was able to specify the expected behavior within my tests.
- For this project, when I handled unexpected inputs I used console.warn and console.error to indicate what was happening in the code. I feel like it was excessive and next time I do a similar project, I plan to not approach error handling in this way.
- When implementing the actual pathfinding code, I was able to write out the BFS and DFS just fine; however, since the implementation of the A* required a priority queue, I had to write my own implementation of one. It is not optimal and by product, I named the file holding the code as "ScuffedPriorityQueue.js".
- For finishing touches, I added functionality to highlight the expanded nodes so that I could see how the path was calculated. I am glad that I did this because the value function that assigned the value to each node for the A* was wrong. When getting the distance to the goal. I did not apply an absolute value to the differences and by product the A* method would expand to the far corner of the map before exiting and returning the goal.

**Keywords**

A* BFS DFS Javascript Data Structures Computer Science Pathfinding