Post-project Writeup - Maze

James Viner

Project Summary

The task was to create a command-line program that accepts an ASCII maze and prints the shortest possible solution to it, utilizing the graph data structure and Dijkstra's algorithm.

Challenges

Nine hours is a new record for me, in terms of "how long I've spend stuck on a singular problem." I'll admit, this was definitely the most complex problem I've had yet, even with the full implementation of the graph data structure and Dijkstra's algorithm given to me. As usual though, the most complex problems seem to have the dumbest solutions, and what seemed like some completely unrelated bit of code that prevented walls from becoming nodes ended up being the exact piece of code necessary to turn a seg-faulting, memory-leaking pile of garbage into a sleek maze-solver. As it would turn out, having a rudimentary understanding of a data structure doesn't make it that much easier to implement on my own, and I'm glad to have had the libraries you made with us in class.

Successes

The fact that I actually completed the base requirements is a miracle. The above "nine hours" estimate is not hyperbole. I got basic functionality from reading things into graph nodes and then spent nearly 60% of the rest of my time attempting to figure out how to get the algorithm to give me back a list of traversed nodes. That said, when I finally got things working, I was thoroughly satisfied, even if it does feel like my success was a complete 'act of God' fluke. Once base requirements were done, I just knocked out the features I knew would be easy, water and doors, and called it. I wasn't going to spend the additional time rewriting and restructuring things for the wall-breaking feature, and I didn't understand A*.

Additionally, I feel like I had a nice implementation of the actual maze. Storing just the index in each node and using the index as a reference against a stored map of the original maze felt like a neat solution and is marginally more lightweight than using coordinates (as I heard some other people doing). I'm satisfied.

Lessons Learned

That neat trick you showed off for passing a variable as a void * paid off and was used extensively in this project. Similarly, I honestly hadn't thought to structure a 2-dimensional array as a 1-D array that wraps at certain points, but I put it to use on this project as well. It actually made managing the formatting of the maze print at the end pretty easy. Also, there are a lot of ways to implement Dijkstra's algorithm. The same goes for A*, as it would turn out. I understand that there's a thousand ways to program the same thing, but looking up reference material for a particular algorithm can be a mixed bag, based on people's differing implementations. This isn't explicitly good or bad, just a neutral observation.