Ally Smith

Maze Project

1 Modelling

I modeled the maze using two separate graphs, one for forwards traversal and one for backwards. Each time the search lands on a circled node, it switches to the other graph, emulating the rules of Apollo's maze with Diana. Additionally, each node connects to the opposite color of nodes within the path of its direction. For example, a blue node facing northeast would connect to any red nodes along the NE diagonal from the original node in the forwards graph. In the backwards graph, the connections would be made with red nodes in the SW direction. Below is a small maze + the corresponding graphs:

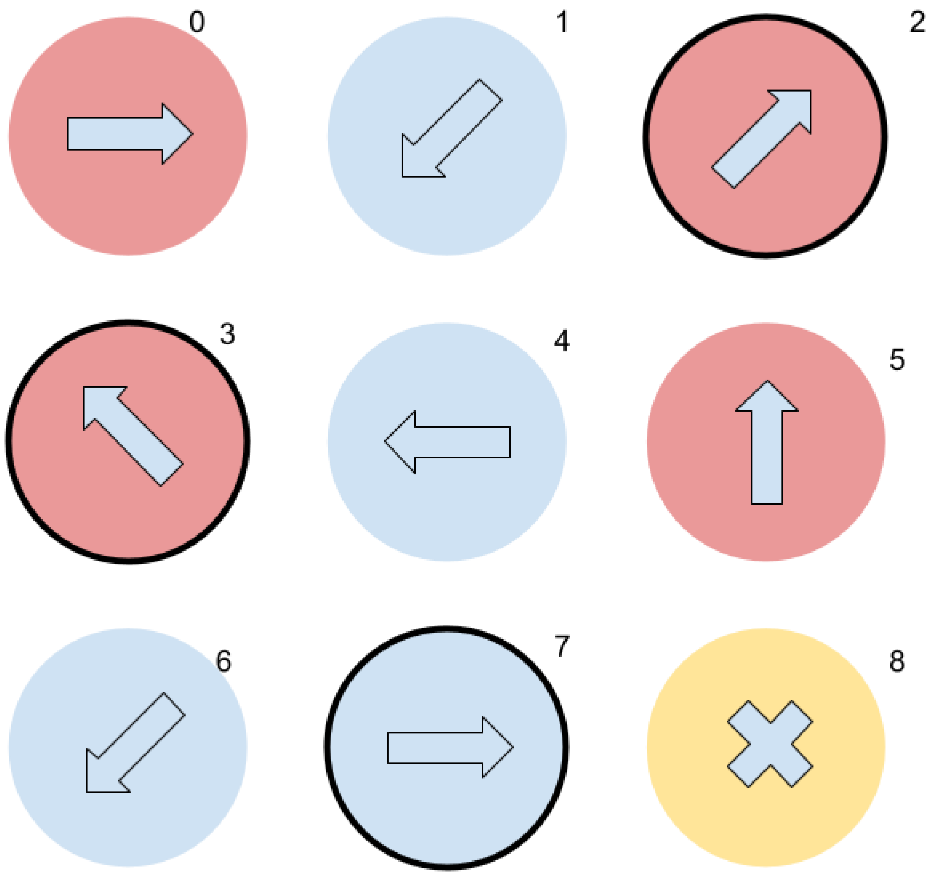

Figure 1: Example maze with a few circled nodes, each labeled with their index.

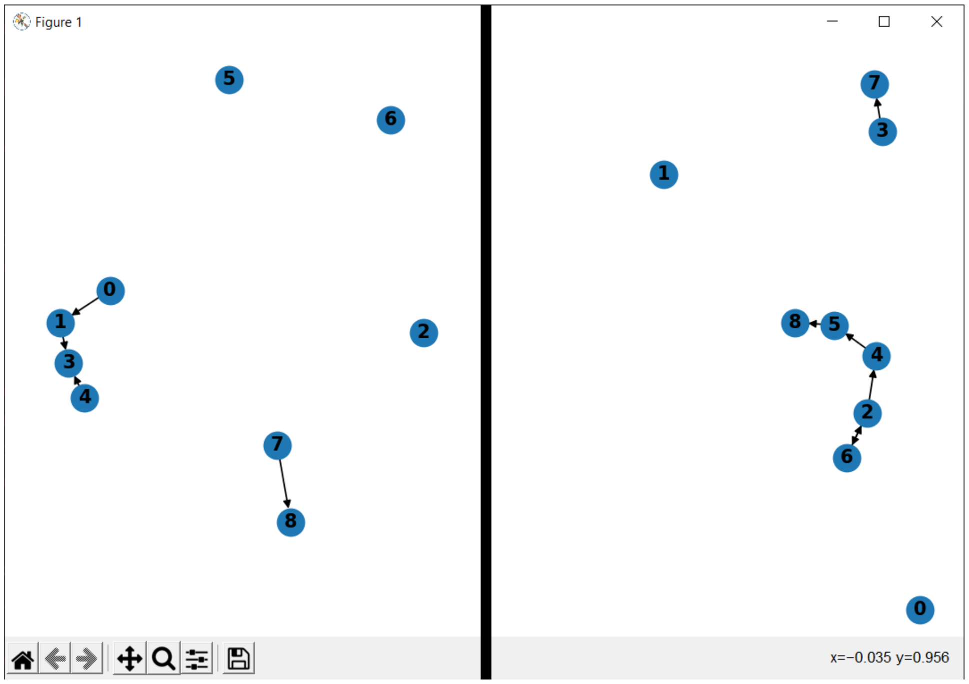

Figure 2: The graphs generated by the Python package *networkx*. On the left is the forward-moving edges and the right backwards.

Using this model for the problem, we can apply a breadth-first search algorithm to find the shortest path through the maze. We can convince ourselves this is the case by imagining these graphs were connected together between circled nodes in the two graphs. Therefore, each time the BFS gets to a circled node, it traverses the edge between the graph, effectively switching between the circles and moving backwards, and the regular forward motion. The algorithm will find a path if one exists within the maze by exhaustively searching each node but ensuring that nodes are not repeated.

2 Code

3 Results