

This program makes use of BFS(Breadth first search) and DFS(Depth First search) in-order to solve the maze.

The program uses the following concept:

- 1) It uses either BFS or DFS to check all the next actions it can perform.
- 2) The actions include moving: up, left, down, right.
- 3) After each action it checks if it has reached its goal.
- 4) If not then it explores the next action.

The main difference between BFS and DFS is the fact that BFS ensures it checks all the paths near it first. While, DFS chooses one at random and performs the actions until it reaches a dead end. When solving puzzles where the goal is far away from the start point using DFS is a good strategy as it explores less possibilities. In other cases BFS will ensure a better solution to the problem.

However, if you like a bit of a challenge you could implement the use of "COST". You could number each square in the puzzle with a number. Higher the number greater the cost of moving towards the number. This will help the AI get a very efficient solution to the puzzle. It could choose a single path at a conjunction like the DFS. But it will not be a random selection instead the AI would see which action would reduce the cost and take that path. My idea of the "cost" would be to use the number of squares to travel to reach the goal.

I hope you learn something new from this! Keep coding!