ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

Surname : **MAHASHA**

Name : **KGAUGELO**

Student Number : **218038561**

Module Name : **Artificial Intelligence (IT08X97)**

TOPIC : MAZE TRAVERSAL PROBLEM

Table of Contents

1. Background

1.1 Problem statement

A maze is an interesting traversal puzzle that consists of confusing interconnected paths and obstacles. An Artificial intelligent agent must find a route from the starting point to the destination. The Artificial intelligent agent finding a path from the starting point to the endpoint is called maze solving.

For many years now, pathfinding has been an area of interest in computer science. The maze traversal problem is one of the most popular but very difficult problems for artificial intelligence to find the shortest path to exit the maze. The maze is made up of a 2d grid where some cells in the grid are obstacles, and the Artificial intelligent agent must avoid the obstacles while moving towards the maze's exit. The goal for the Artificial intelligent agent is to reach the maze's exit in as few steps as possible, which will then be the shortest path from start to exit.

The problem of finding the shortest path to exit a maze must be solved in real-time, and there are limited computational resources like CPU and memory. The traditional method for maze solving Artificial intelligent agents is based on trial-and-error methods. The agents try each available path in the maze until finding a path that leads to the destination.

1.2 Proposed solutions

The traditional trial and error way of solving the maze can be resource-heavy and time-consuming. The trial-and-error method does not give the shortest path as it tries all the possible paths. One of the ways to avoid the problems mentioned above is to use Graph theory algorithms as they are one of the most efficient ways to analyze and find the shortest path to the destination. Many Graph theory algorithms can be used to implement an Artificial intelligent agent that can solve the maze traversal problem. One of the most popular algorithms for solving this problem is the Best first search, A*, and breadth-first search algorithms.

In this assignment, the maze will be represented as a 2D grid. The size of the grid is unbounded. The 2D grid will have obstacles on some cells, and the agent can only go around the cells. It cannot go through the cells. The Artificial intelligent agent will be tasked with finding the shortest path to traverse the 2D grid from the starting point to the exit of the grid. For this assignment, the Artificial intelligent agent will use a Graph theory approach to solve the maze traversal problem. The Artificial intelligent agent will use a Breadth-First search approach to search for a possible path to the solution. Breadth-First search will always find the shortest path to the solution. Breadth-first search guarantees the shortest path, but it is resource-heavy as it is not using a heuristic search approach to search the graph. In this assignment, I will be focusing on getting the shortest path to solving the problem, not necessarily the performance of the algorithms.

2. Classification of Agent

2.1 The structure of Agents

A single agent will be responsible for traversing the maze and finding the shortest path for this assignment. The agent can be classified as a simple reflex agent as it will select actions based on its current position on the grid; it will not consider the percept history. The agent can also be classified as a Goal-based agent as in every state, and the agent can use a heuristic function to measure how far it is from reaching the goal. It will use the heuristic function to expand nodes with a higher potential to reach the goal.

The agent can learn because initially, It operates in an unknown environment with only the given rules. The agent then learns and becomes more competent than the initial knowledge it had before entering the maze.

2.2 Properties of task environments

The properties of the task environment are as follows,

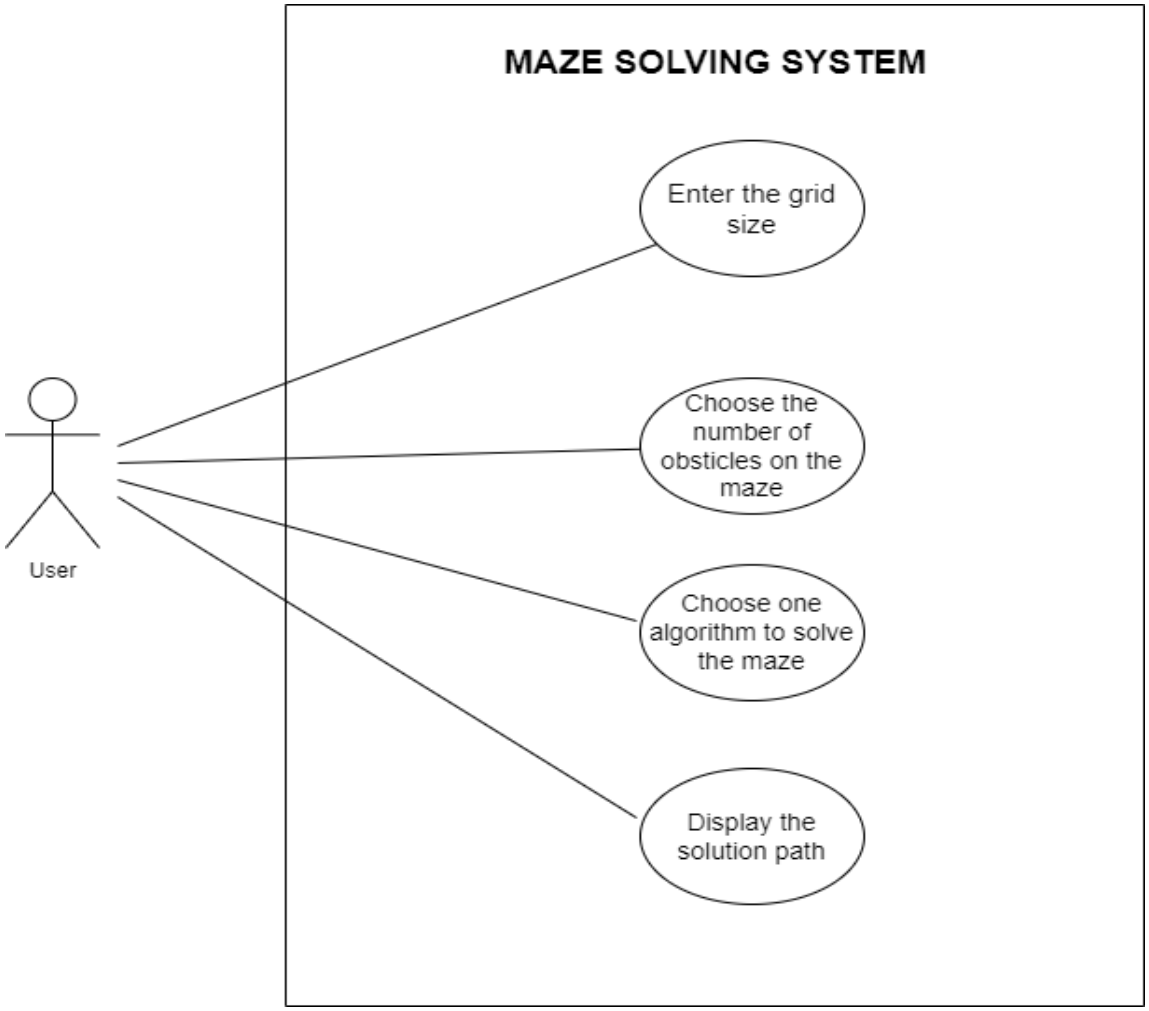
- **Fully observable** – At every point on the grid, the agent has access to the complete state of the environment.
- **Agents** – The agent is a Single-Agent.
- **Non-deterministic** – The next state of the environment does not depend on the actions executed by the agent in the current state.
- **Episodic** – The agent receives a percept then performs a single action based on the percept. The agent can only move left, up, down, or right at any given state, depending on the location on the grid. In some locations on the grid, the agent has a limited number of available moves.
- **Static environment** – The environment does not change while the agent travels through the maze.
- **Discrete** – In any environment, the agent has a finite number of distinctive paths the agent can take to solve the maze.
- **Know environment** – The outcome of every action is known

2.3 PAES description

Agent type	Performance measure	Environment	Actuators	Sensors
Maze traversal Single-agent	Avoid obstacles, minimize the number of steps, move towards the exit after every move	Maze (2d grid), walls blocking some cells, exit cell, starting cell		

3. Use Cases and Use Case Diagrams

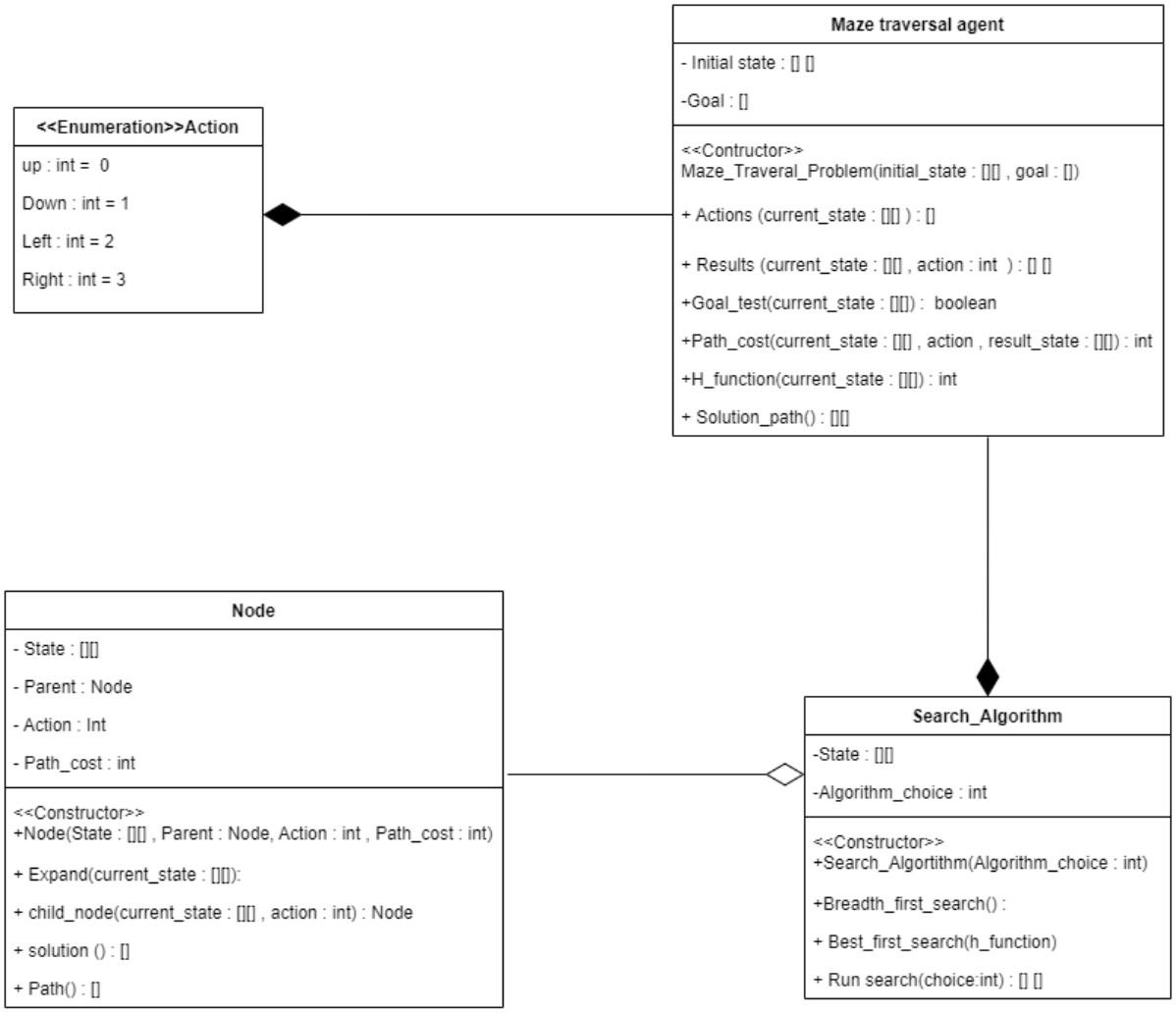
3.1 Use Case Diagram



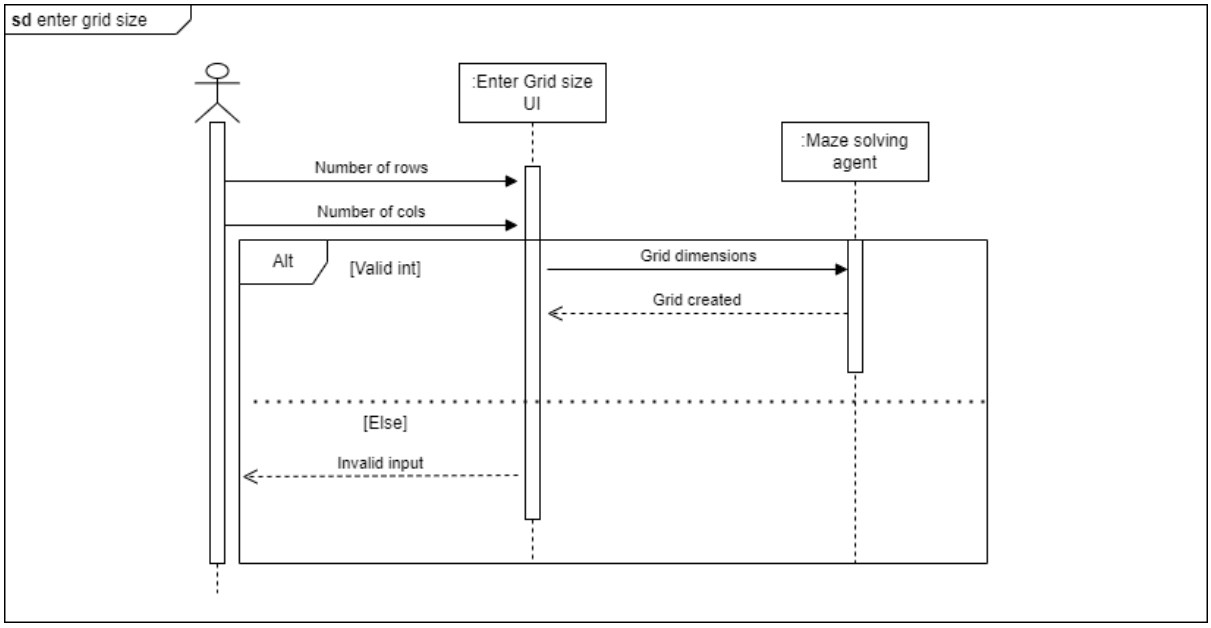
3.2 Use Case descriptions

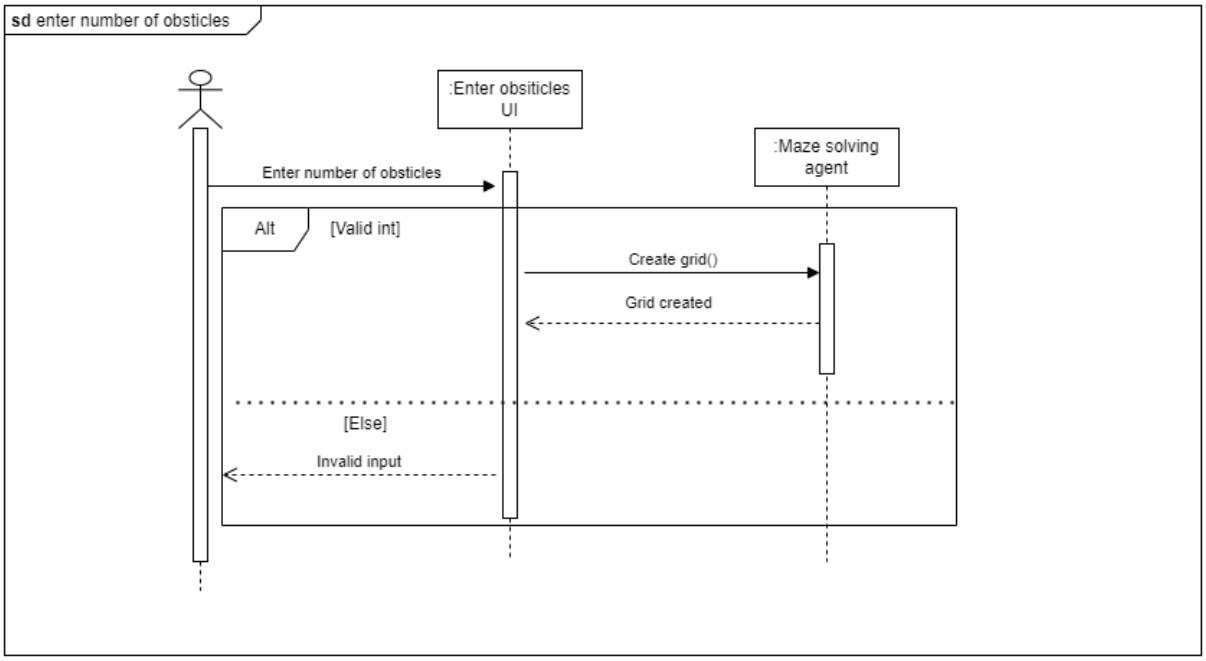
Use Case	Actor	System	Action
Enter the grid size	User	Maze solving system	1. Enter the number of rows 2. Enter the number on columns 3. The system creates a grid with the specified dimensions
Enter the number of obstacles on the grid	User	Maze solving system	1. Enter how many cells on the grid should have obstacles 2. The system populates the grid with obstacles
Choose an algorithm to solve the maze	User	Maze solving system	1. Choose between Breadth-first search algorithm and A* search algorithm The system solves the maze with the selected algorithm
Display the solution path	User	Maze solving system	1. Click display solution 2. The system displays the shortest path to solve the maze

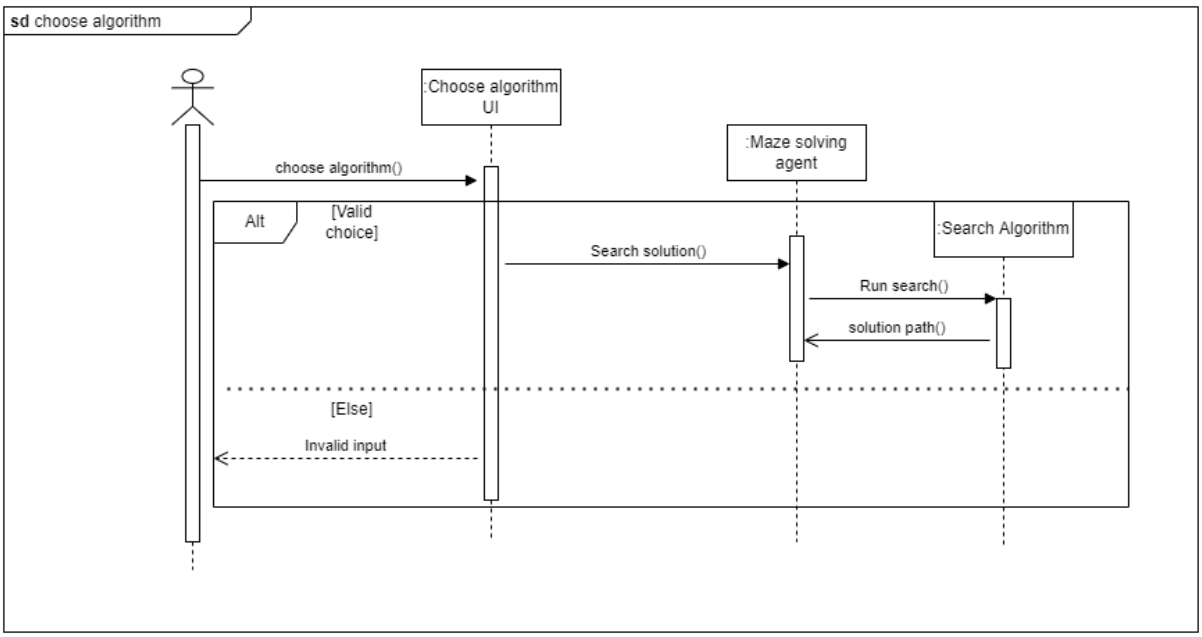
4. Class Diagrams

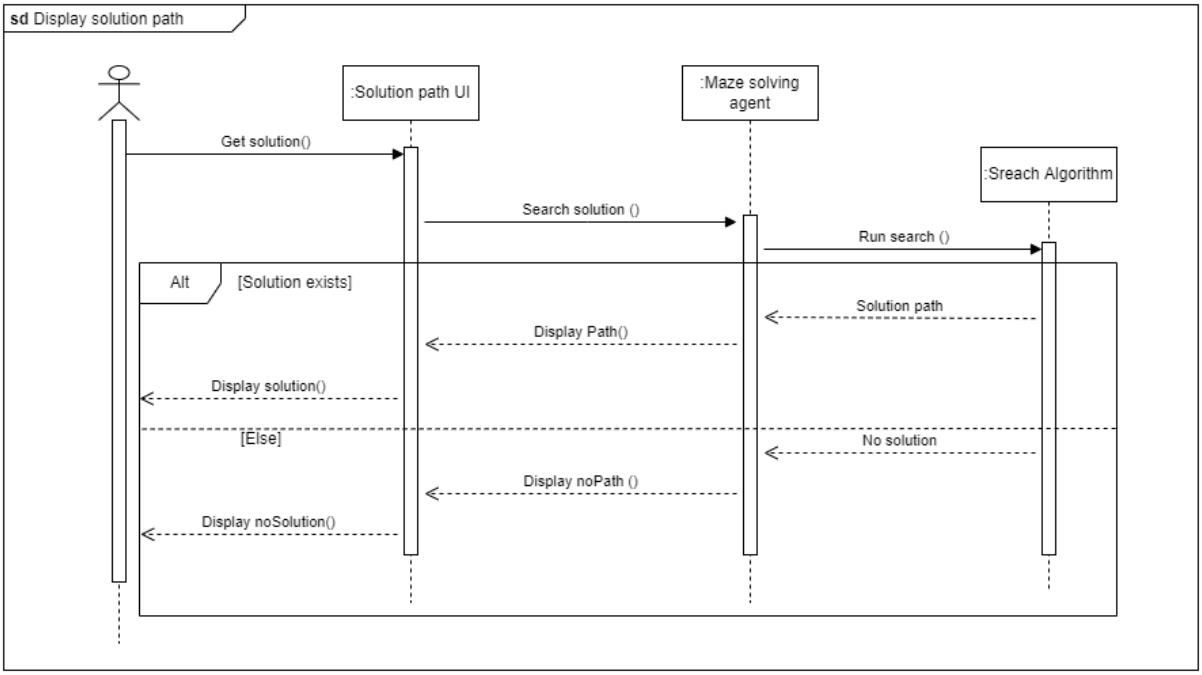


5. Interaction Sequence diagrams






sd choose algorithm
:Choose algorithm UI
:Maze solving agent
choose algorithm()
Alt
[Valid choice]
:Search Algorithm
Search solution()
Run search()
solution path()
[Else]
Invalid input


sd Display solution path
:Solution path UI
:Maze solving agent
Get solution()
:Sreach Algorithm
Search solution ()
Run search ()
Alt
[Solution exists]
Solution path
Display Path()
Display solution()
[Else]
No solution
Display noPath ()
Display noSolution()

6. Intelligence of agent

A maze-solving artificial intelligent agent with an intelligent approach based on graph theory algorithms was introduced. The graph theory algorithms that the agent uses are breadth-first search and best-first search algorithms. A comparison of the two algorithms will be made, and it will be based on the time it takes each algorithm to solve the maze and the length of the solution path. Therefore the final program will be developed based on the most efficient algorithm.

The agent will demonstrate intelligence in a way that it will explore possible paths and choose the path that leads to a solution, which will be the shortest path. The agent should be able to use the set of rules provided and then navigate any maze and avoid the obstacles on the maze.

The artificial agent's intelligence will be tested by changing the maze and adding more obstacles to make the problem more difficult. The agent should be able to solve the problem if there is a solution. Otherwise, it should also be able to detect if the maze does not have a solution.

References

1. Barnouti, N.H., Al-Dabbagh, S.S.M. and Naser, M.A.S., 2016. Pathfinding in strategy games and maze solving using A* search algorithm. *Journal of Computer and Communications*, *4*(11), p.15.

2. Aqel, M.O., Issa, A., Khdair, M., ElHabbash, M., AbuBaker, M. and Massoud, M., 2017, October. Intelligent maze solving robot based on image processing and graph theory algorithms. In *2017 International Conference on Promising Electronic Technologies (ICPET)* (pp. 48-53). IEEE.

3. Pathfinding search algorithms, https://www.scirp.org/html/2-1730422_70460.htm?pagespeed=noscript#, last accessed July 25, 2016