CS1001 Advanced Algorithms
Dr. Ali Sadiq

## Project

### Task Summary

In this assessment, you are required to program an Algorithm-based agent to solve a real-world, which is a challenging case study. This assessment is done individually, and you are required to submit programs and supporting documents in report format. Please refer to the Task Instructions for details on how to complete this task. This assessment is intended to test:

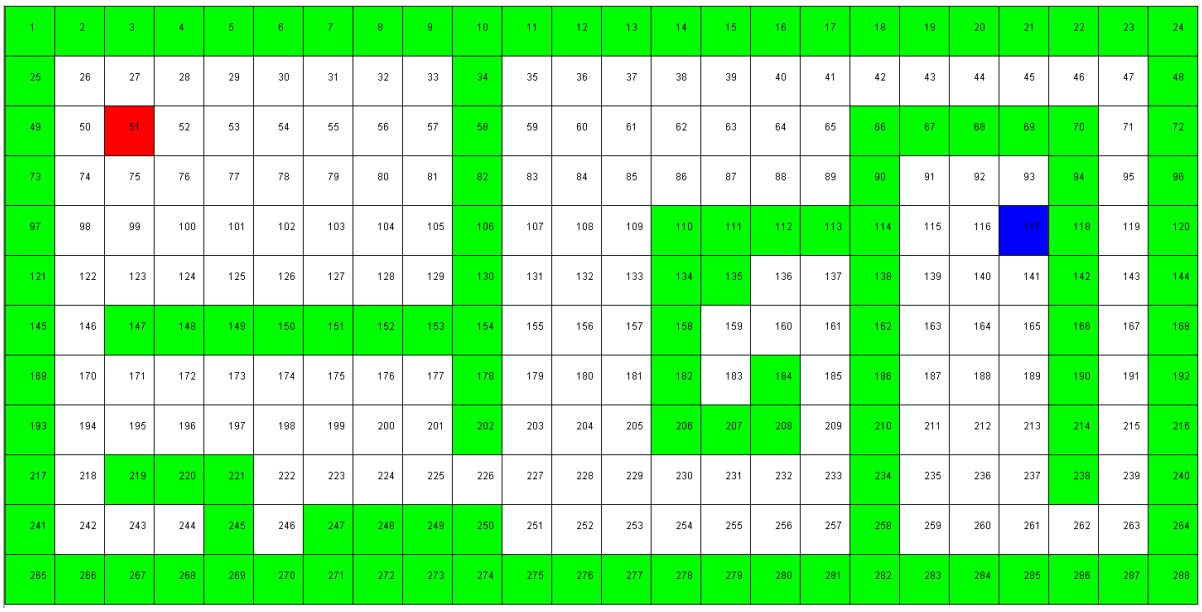
- Your understanding of the theories covered in the course.
- Your ability to formulate and frame a simplified real-world problem for an algorithm as problem solving technique.
- Your ability to choose a suitable algorithm for solving the problem.
- Your ability to implement algorithm as problem solving technique in a modern programming language.
- Your ability to provide a document to discuss the potential applications and ethics of the algorithmic solution.

### Context

You are required to create a robot path planner that is able to find an optimal path to navigate an environment and reach a target. By completing this assessment, you will show your skills on leveraging the best algorithm to solve a simplified real-world problem.

The maze can be seen in the image below. It can be seen that there are 12 rows and 24 columns, meaning there is a total of 288 blocks on the map. There are four different types of blocks in this map as follows:

- Green: wall
- White: space (void)
- Red: initial position of the robot
- Blue: the target



In C++, you can easily represent the entire maze using the following 2D array:

```
int maze[12][24] =
            {
                  {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},
                  {1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1},
                  {1,0,-1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,1,1,1,1,0,1},
                  {1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,1},
                  {1,0,0,0,0,0,0,0,0,0,1,0,0,0,1,1,1,1,1,0,0,9,1,0,1},
                  {1,0,0,0,0,0,0,0,0,0,1,0,0,0,1,1,0,0,1,0,0,0,1,0,1},
                  {1,0,1,1,1,1,1,1,1,1,1,0,0,0,1,0,0,0,1,0,0,0,1,0,1},
                  {1,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,1,0,1,0,0,0,1,0,1},
                  {1,0,0,0,0,0,0,0,0,0,1,0,0,0,1,1,1,0,1,0,0,0,1,0,1},
                  {1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,1},
                  {1,0,0,0,1,0,1,1,1,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1},
                  {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},
            };
```

In this array, we use the following numbers to represent different types of blocks:

- 1: wall
- 0: space (void)
- -1: initial position of the robot
- 9: the target

In this assessment, you need to design and implement a robot path planner. Note you could select any of the taught algorithms in this course to be as your searching algorithm that you have to implement for robotics' path planner. You do not have to draw a maze like the picture above. You can simply mark the shortest path obtained by your algorithm using for instance 2 (or any other numbers except 1, 0, -1, and 9) in the 2D array. If you draw the maze, however, it will be a lot easier to visualise the path and make sure that it is the closest path to the target when you are testing your program.

After implementation, write a reflective report detailing the experience of the development process. The report should be at least 1500 words and not more than 15 pages in length and include the following sections:

- Overview
- Review on some of the related robotics planners and utilised techniques
- Your designed planner and the flowchart of the development stages, this involving:
    - ✓ What went right
    - ✓ What went wrong
    - ✓ What you are not sure about
- Performance analysis that involves computational complexity of the proposed planner strategy and computational actual average time taken during the running time.
- Conclusion
- List of references in Harvard format.

## Task Instructions

- Appropriate, effective, and correct usage of C++ or Python.

- Good selection of search algorithm.
- Effective use of search algorithm.

The source code that you will be submitting:

1. Should be free or build warning, build errors, and all intermediate files (.obj, .pdb, etc), crashes, and errors (compile, run-time, logical, etc.).
2. Your code should be structured and written with the best practices in the field of programming.
3. There should be enough number of comments in the source files to show your understanding of the program.
4. Any third-party code should be appropriately attributed.

**When you submit the electronic version of your project make sure to use the following names:**

- Name the source code folder as: Source – Student Name

## Submission Instructions

Please submit a ZIP file including:

- **Release Build Zip:** A release build executable must be zipped and included with the submission. Ensure that project settings are set to Release when creating this build.
- **Source Code Zip:** All relevant source code files and project files must be zipped and included with the submission.
- **Reflective report:** PDF or Word
- Naming & File structure for the zip file.
    - CourseCode_Project_LastName_FirstName.zip
        - Project_Build_LastName_FirstName.zip
        - Project_Source_LastName_FirstName.zip
        - Project_report_LastName_Firstname.pdf
        - Project_ report _LastName_Firstname.docx

## Assessment Rubric

| Assessment Attributes | Fail (Yet to achieve minimum standard) 0-49% | Pass (Functional) 50-64% | Credit (Proficient) 65-74% | Distinction (Advanced) 75-84% | High Distinction (Exceptional) 85-100% |
|---|---|---|---|---|---|
| Work demonstrates the knowledge and understanding of the best knowledge representation methods for the case study considered in the assessment<br><br>40% | Little or no knowledge of the best knowledge representation methods for the case study considered in the assessment. A state space tree or other standard algorithmic representation methods are not used. | Acceptable but further work is required to show the knowledge of the best knowledge representation methods for the case study considered in the assessment. A state space tree or other standard algorithmic representation methods are used but include errors and flaws. | Good level of knowledge about the best knowledge representation methods for the case study considered in the assessment. A state space tree or other standard algorithmic representation methods are used but not in an efficient manner for the problem. | Excellent but not thorough knowledge about the best knowledge representation methods for the case study considered in the assessment. A state space tree or other standard algorithmic representation methods are used but it is not robust and error free for different mazes. | Excellent and thorough understanding of the best knowledge representation methods for the case study considered in the assessment |
| Work demonstrates the knowledge and understanding of the search algorithm for the case study considered in the assessment<br><br>40% | Little or no knowledge of the search algorithms for the case study considered in the assessment. The search method is attempted but not implemented correctly. | Acceptable but further work is required to show the knowledge of the search algorithms for the case study considered in the assessment. The search method is implemented but includes errors and flaws. | Good level of knowledge about the search algorithms for the case study considered in the assessment. The search method is implemented but not in the most efficient manner. | Excellent but not thorough knowledge about the search algorithms for the case study considered in the assessment. The search method is efficient, but it is not robust and error free for different mazes. | Excellent and thorough understanding of the search algorithms for the case study considered in the assessment. The search method is highly efficient, robust, and error free. |
| The reflective essay demonstrates the knowledge and understanding of the whole process of problem solving using the best algorithmic problem-solving methods and practices<br><br>20% | The reflective essay includes no, or little sections and concepts required. There is no or little elaborations or justifications. | The reflective essay includes some of the sections and concepts required. There is little elaborations or justifications to demonstrate the knowledge and understanding of the whole process of problem solving using the best algorithmic problem-solving methods and practices. | The reflective essay includes all the sections and concepts required. Elaborations and justifications are not discussed well to show the mastery of the algorithmic technique used to solve the problem. | The reflective essay includes all the sections and concepts required. Elaborations and justifications are not thorough and in-depth to demonstrate the knowledge and understanding of the whole process of problem solving using the best algorithmic problem-solving methods and practices. | The reflective essay includes all the sections and concepts required. Elaborations and justifications are thorough and show the mastery of the process of solving a simplified real-world problem using an algorithmic technique. |