# Project team

|  |  |  |  |
| --- | --- | --- | --- |
| Team leader | [20P3844] | [Youssef Emad] | [20P3844@eng.asu.edu.eg] |
| Team member | [18P7231] | [Hamza Abdellah] | [18P7231@eng.asu.edu.eg] |
| Team member | [18P8121] | [Basel Mohamed] | [18P8121@eng.asu.edu.eg] |
| Team member | [18P9203] | [Mostafa Essam] | [18P9203@eng.asu.edu.eg] |
| Team member | [20P8736] | [Heba Hesham] | [20P8736@eng.asu.edu.eg] |

# Background

## Why is the project being undertaken?

Entertainment: The project has been undertaken as part of an effort to develop a fun and challenging game or puzzle that involves maze-solving. The maze solver application can be used as a tool to generate and solve these mazes, providing players with an interesting and engaging experience.

## Describe an opportunity or problem that the project is to address.

* The maze solver application can be used to solve a variety of maze-related problems, such as finding the shortest path through a maze, navigating through a maze, or testing the efficiency of maze-solving algorithms. It can also be used for educational purposes to teach students about graph theory and search algorithms such as BFS.
* One specific opportunity for the maze solver application is in the field of robotics. Robots are often used in environments that require navigation through complex mazes, such as in search and rescue operations, exploration of unknown territories, or inspection of hazardous areas. The maze solver application can be used to test and optimize the navigation algorithms of these robots, ensuring that they can efficiently and safely navigate through the maze-like environments they are designed to operate in.
* Another potential use case for the maze solver application is in game development. Maze-solving algorithms can be used to generate procedural mazes in games, which can provide players with a challenging and unpredictable experience. The maze solver application can be used to test and optimize these algorithms, ensuring that the generated mazes are interesting and solvable.

# Objectives

* Developing efficient maze-solving algorithms: One of the primary objectives of a maze solver project is to develop efficient algorithms that can solve mazes quickly and accurately. This may involve exploring different search algorithms such as Breadth-First Search (BFS), Depth-First Search (DFS), or A\* search, and optimizing these algorithms to improve their performance.
* Generating solvable mazes: Another objective of a maze solver project is generating mazes that can be solved using the developed algorithms. This may involve developing algorithms that can generate random mazes of different sizes and complexities, while ensuring that the mazes are solvable.
* Testing and validating maze-solving algorithms: The maze solver project also be aimed at testing and validating the developed algorithms on different types of mazes. This may involve creating a large database of mazes with varying sizes and complexities, and testing the algorithms on these mazes to ensure that they can solve them efficiently.
* Developing a user-friendly interface: Another objective of a maze solver project may be to develop a user-friendly interface that allows users to input mazes and visualize the maze-solving process. This may involve developing a graphical user interface (GUI) that allows users to input mazes using a mouse or keyboard, and displays the maze-solving process in real-time.

# Scope

1. Algorithm development: The scope of the project will include developing and implementing different maze-solving algorithms, such as BFS, DFS, or A\* search. This may involve exploring different variations of these algorithms and optimizing them for performance.
2. Maze generation: The scope of the project will also include developing algorithms that can generate random mazes of different sizes and complexities. This may involve exploring different maze generation techniques, such as randomized Prim's algorithm or recursive backtracking, and optimizing them to generate mazes that are solvable by the developed maze-solving algorithms.
3. Maze-solving visualization: The scope of the project will also include developing a user-friendly interface that allows users to visualize the maze-solving process. This may involve developing a GUI that displays the maze and the path taken by the algorithm in real-time, as well as providing additional information such as the number of steps taken, and the time required to solve the maze.
4. Testing and validation: The scope of the project will also include testing and validating the developed algorithms on different types of mazes. This may involve creating a large database of mazes with varying sizes and complexities, and testing the algorithms on these mazes to ensure that they can solve them efficiently.

# System overview

1. Maze Solver App
2. Algorithms development
3. Maze Generation
4. Maze-Solving Visualization
5. Testing and Validation

# Timeframe for deliverables

|  |  |  |
| --- | --- | --- |
|  | Description of Work | Start and End Dates |
| Phase One | Research and Planning | 8/4 to 22/4 |
| Phase Two | Development, Implementation and Testing | 23/4 to 7/5 |
| Phase Three | Designing and Developing the GUI | 8/5 to 22/5 |

# Monitoring and Evaluation

1. Performance testing: The performance of the maze solver application can be evaluated by running tests on various mazes of different complexities and sizes. The time taken to solve the maze and the number of steps taken can be recorded and compared to other maze-solving algorithms to determine the efficiency of the application.
2. User feedback: The ease of use and effectiveness of the maze solver application can be evaluated by soliciting feedback from users. Users can be asked to provide feedback on the user interface, the accuracy of the solution, and the overall experience of using the application.

# Approval Signatures

|  |  |  |
| --- | --- | --- |
| Supervisor |  | Teaching Assistance |
| Associate Professor / Walaa H. Elashmawi |  | Eng. Ziad Elgayar |