AI BASED SELF DRIVING CAR

PROJECT PROPOSAL

Course code: AI-2002

Section: 6C

Discipline: BS-CS

GROUP MEMBERS

K21-3398: Laiba Nadeem

K21-4605: Wassay Uddin

K21-4502: Ahmed Mustafa

1. Introduction

The Flag Agent Maze Game is a Python-based project that simulates an environment where agents navigate through a maze to collect flags while avoiding obstacles. The game utilizes various algorithms such as Breadth-First Search (BFS), Depth-First Search (DFS), A Search, and Hill Climbing to demonstrate different pathfinding strategies.*

2. Objective

The primary objective of the project is to develop an interactive maze game where multiple agents employ different algorithms to reach designated flags while optimizing their paths based on specific heuristics.

3. Features

- *Dynamic Maze Generation: The maze is dynamically generated with random placement of coins, potholes, barriers, and flags.*
- *Multiple Algorithms: Agents can utilize different algorithms including BFS, DFS, A*, and Hill Climbing to navigate the maze.*
- *Heuristic Selection: For A* and Hill Climbing algorithms, users can select from multiple heuristic functions such as Manhattan distance, Euclidean distance, and Chebyshev distance.*
- *Agent Interaction: Agents move within the maze, collecting coins, avoiding potholes, and reaching flags based on their selected algorithms.*
- *GUI Interface: The project provides a graphical user interface (GUI) for users to input the number of flags, agents, flag coordinates, agent starting coordinates, and algorithm selection.*

4. Implementation

- *Maze Class: Manages the maze grid, generates the maze with random elements, handles agent interactions, and displays the maze.*
- *Agent Class: Represents the agents navigating through the maze. Implements various algorithms for movement, including BFS, DFS, A*, and Hill Climbing.*
- *FlagAgentGUI Class: Provides a GUI interface for users to input game parameters and coordinates for flags and agents.*

5. Algorithms Used

- *Breadth-First Search (BFS): Explores the maze level by level, ensuring the shortest path to the flag.*
- *Depth-First Search (DFS): Explores the maze depth-wise, prioritizing exploration of one branch before moving to the next.*
- *A* Search: Utilizes a heuristic function to estimate the cost of reaching the flag, guiding the agent towards the goal while considering the shortest path.*

- *Hill Climbing: Iteratively improves the agent's position by selecting neighboring cells with lower heuristic costs until reaching the flag.*

6. Future Enhancements

- *Improved Maze Generation: Implement algorithms for maze generation such as Prim's algorithm or Recursive Division for more complex and diverse mazes.*
- *Additional Heuristic Functions: Introduce more heuristic functions to enhance the effectiveness of A* and Hill Climbing algorithms..*
- *Enhanced GUI: Enhance the GUI with more interactive features such as real-time visualization of agent movements and maze dynamics.*

7. Conclusion

The Flag Agent Maze Game project offers an engaging platform to explore various pathfinding algorithms in a simulated maze environment. By allowing users to interactively control the parameters and algorithms, the project serves as an educational tool for understanding different search strategies and their applications in real-world scenarios.