كلية الحاسبات والمعلومات
جامعة المنصورة

Artificial Intelligence

Course Project

Rescue Mission

Prepared by

Eng / Mariam Ahmed

Eng / Abdullah Samir

I. Project Description

The project simulates a hostage rescue operation using AI techniques.

The AI agent uses the A* pathfinding algorithm to navigate the building efficiently, avoid obstacles and reach the hostages in the shortest possible way.

The simulation is implemented using the Unity Engine for the environment and graphics and C# for scripting the logic and algorithms.

The scenario includes:

- A 2D map representing the building layout (walls, rooms, obstacles).
- Hostages located at specific points.
- The AI agent plans the best path to reach the hostages and avoid obstacles.

II. Project Objectives

- Show how AI decision-making can optimize time-critical operations.
- Create an interactive simulation that represents the planning and execution of a rescue.

- Gain practical experience in combining AI algorithms with game engines like Unity.

III. Agent Type

Goal-based Agent, as the agent makes its decisions based on a specific goal (reaching and rescuing the hostages), not just on the current state

IV. ODESDA Model

- Observable => Partially Observable: The rescue agent can't see the entire map initially (it discovers obstacles while moving).
- Deterministic => Deterministic: Each action leads to a predictable and fixed outcome.
- Episodic => Sequential: Current actions affect future states, each decision impacts later stages of the mission.
- Static => Static: The environment doesn't change while the agent is moving.
- Discrete => Discrete: The map is divided into cells, and the agent moves from one cell to another.

- Agent => Single Agent: Only one rescue agent operates to achieve the mission.

V. PEAS Model

- Performance Measure: number of hostages successfully rescued, and avoidance of static and dynamic obstacles.
- Environment: a 2D map of a building with fixed obstacles (walls), dynamic obstacles (enemies) and static hostage locations.
- Actuators: movement controls (move forward, move backward, turn left, turn right, stop) and a rescue action triggered upon reaching a hostage.
- Sensors: distance sensors to detect nearby obstacles and location detection to identify hostage and enemies locations.

VI. Problem Formulation

- Initial State: the rescue agent starts at a specific location on the map.

- Successor Function: the agent can move up, down, left or right.

- Goal Test: the agent reaches the location of a hostage and performs the rescue action.
- Path Cost: 1 per move from one cell to another.

VII. Tools and Technologies

- Unity Engine: create the simulation environment, design the map layout and handle the visual representation of the agent and its movements.
- C# Programming Language: implement the logic of the agent, the A* algorithm and the interaction between the agent and the environment.
- A* Algorithm: calculate the optimal path between the agent's current location and the hostages while avoiding obstacles and minimizing the travel cost.

VIII. Algorithm (A* Pathfinding)

- Each cell in the environment is treated as a node.

- The agent explores neighboring nodes based on the smallest f(n) value, where: f(n) = g(n) + h(n)

A. g(n) = cost from the start node to the current node.

B. h(n) = cost from the current node to the goal.

- The agent selects the path with the smallest f(n) to reach all hostages.

IX. System Behavior

- The agent is required to complete the rescue mission within a predefined time limit.
- Collision with an enemy or failure to reach all hostages before the timer runs out will trigger a "Game Over" event.

Thanks For Your Efforts Dr / Sara ElMetwally