Elevator agent

1 Introduction

- I will implement an elevator agent which takes requests for floors to go to and takes the user to those floors
- The agent will perceive the environment and make choices of which floor to go to and if it should open or close its doors based on the perceived environment
- The goal is to take people to their desired floors

2 Agent Design

- Agent type
    - I chose a simple reflex agent for the elevator as an elevator can be implemented without maintaining memory
    - In order for this agent to act without maintaining memory the environment will provide the elevator with some facts at each floor such as a signal whether or not someone at the current floor pressed the up or down button and a signal indicating whether or not someone in the elevator pressed the current floor as their destination
- Percept function
    - The agent perceives if a request has been made for the current floor
    - The agent perceives if it is at the top/bottom of the building
    - Percepts: {at_top, request_for_current_floor, current_direction}
- Condition-Action Rules:
    - The agent follows these rules:
        - If the agent is at the top or bottom floor it switches directions
        - If the agent is at a floor and there's a request for that floor the elevator stops and opens its doors waits and then closes its doors and carries on in the same direction it was traveling in

Environment

- Environment Type:
    - The environment is **partially observable** as the agent can only view if a floor was requested once it is on that floor
    - The environment is **deterministic** since the agent's actions lead to predictable results.
    - The environment is **dynamic** as people can request the elevator and therefor change the environment while the agent is acting
    - The environment is **discrete**, an agent cannot be “between” floors, we have a limited amount of floors bound by the top and bottom floor
- Challenges of the Environment:
    - The main challenge is that people can request to be taken to their floor while the agent is acting. For example we could be on floor 2 traveling up to floor 10 to take someone there and then someone on floor 3 asks to be taken to floor 1. This will result in that persons travel time to be greatly increased with our current implementation of the agent

4 Agent’s Interaction with the Environment

- Sensors
    - The elevator has the sensors to perceive:
        - What direction it is currently traveling in
        - If it is at the top/bottom floor
        - If someone requested the current floor

- Actuators
    - The agent can perform the following actions:
        - Move up/down a floor
        - Stop and let someone onto the elevator
        - Change direction when reached the top/bottom floor

Pseudo Code

If current floor is requested

    Stop and open the door to let the user on the elevator

Else continue in current direction

6 Design Rationale and Justification

- Chosen Environment:
    - We have a dictionary where the keys are in the range from 0 to max_floor, this represents the floors in the building. Each key corresponds to a boolean value representing if someone has requested that floor
    - Note that even though we have a dictionary of all floors our agent can only perceive the floor it is currently on
- Percept Choices:
    - The agent can only observe the current floor it is on and its direction. This aligns with the reflex agent's goal of making immediate decisions based on current input
- Action Rules:
    - The reflex action of "stop on floor and move” if the current floor is requested or “continue to move” if not directly aligns with the agent’s purpose.
    - Since the elevator is always moving up and down we will always cover the full range of floors ensuring that the users will always be collected
- Why Simple Reflex?:
    - A simple reflex agent is appropriate here because the task is straightforward:Collect all people who request to be picked up on their floor. Since we have relaxed the elevators goal from taking people to their destination in the most efficient way to simply ensuring that everyone gets to their destination regardless of how long it takes then there is no need for memory (model-based) or planning (goal-based), as the task doesn’t involve any complex future-oriented decisions

7 Limitations and Improvements

- Limitations
    - The simple reflex agent cannot plan ahead and ensure it moves efficiently between floors. Although everyone will eventually get to their desired floor it may not be in the fastest, most efficient way
- Improvements
    - If the agent could access the state and see which other floors are requested besides the one it is currently on it could plan more efficient paths to floors which will reduce the time it takes to stop at all required floors.

8 conclusion

- The simple reflex agent successfully takes every user to their requested floor but it will not do this in the fastest way and may result in people who enter close to their requested floor to still wait long periods of time eg: if the elevator is traveling up and you get on at floor 2 and want to go to floor 1 you have to wait for it to travel to the top and then all the way back down
- If we create a more sophisticated agent architecture and allow the agent to access all requests we can plan the movements of the elevator resulting in better performance