MEMBERS

Jonathan Thom : BIT/21/SS/030

Bischof Nkhata : BIT/21/SS/028

Peter Zimba : BIT/21/SS/031

Introduction

This project implements an AI-based Vacuum Cleaner Robot that autonomously moves between rooms to clean them. The robot follows a predefined logic to check the state of a room (dirty or clean), clean (sucks) it if necessary, and then move to the next room.

The project is divided into two tasks:

1. Task 1: Implementation of a simple vacuum cleaner in a two-room environment.
2. Task 2: Extension of the implementation to multiple rooms, floors, and additional constraints.

Task 1: Simple Vacuum Cleaner Robot

The objective of Task 1 was to simulate a Vacuum Cleaner AI Agent that operates in a simple environment consisting of two rooms. The agent detects whether a room is dirty or clean and acts accordingly.

1.1 Is the Robot Rational?

Yes, it is rational because it manages to;

- Perceive its environment (checks if a room is dirty).
- Take an appropriate action (cleans the dirty room).
- Move out of the room if it finishes cleaning.

1.2 Implementation Approach

Two levels have been implemented.

1. Easy: The agent is initialized with a predefined environment state.

    - A codespace link for the easy task program **https://bug-free-space-palm-tree-vx97769j6jx266q.github.dev/**

2. Hard: The agent is given random inputs from a text file. The text file has about 300 lines. Data is getting extracted from the text file from each line to the agent.

- A codespace link for the Hard task program https://bug-free-space-palm-tree-vx97769j6jx266q.github.dev/

1.5 Results

- The agent is able to check if the room is dirty or not.
- The robot successfully cleans the dirty room and skips already clean rooms.
- The robot moves between two rooms efficiently.

Task 2: Multi-Room & Multi-Floor Vacuum Cleaner Robot

This task extends the Vacuum Cleaner AI to handle multiple rooms, floors, and additional constraints:

1. A corridor-based movement system where the robot follows a structured path.
2. Time-based where the agent records the time it takes to clean a specific room.
3. Multiple floors and stairs.
4. Dynamic room availability, where some rooms are occupied or unavailable.
5. A learning component, allowing the robot to tell which rooms are mostly likely to get dirty for the day.

2.2 Environment Representation

A dictionary-based building representation is used.

The dictionary has another dictionary in it where each it has a key is a room and each value represents dirty.

If the value is empty, it means the room is clean and if the value has a star in it means the room is dirty.

The more the number of stars the dirtier the room and the more the time it takes to clean the room.

2.3 Implementation Approach

1. The building has been represented as a nested dictionary where each inside dictionary represents a floor containing rooms having their corresponding values.
2. Two implementations have been made, each addressing some of the goals listed

    - A hardcoded dictionary having two options; empty, and stars where empty means its clean and stars means it dirty. The more the number of stars the dirtier the room is
    - A randomly generated status of the room bearing a number from 0 to 10 where 0 means its clean and any number greater than 0 means its dirty. The greater the number the dirtier the room is.

3. Time-based simulation is added to track cleaning duration.
4. Room occupation status (free, occupied, or makeup class) is introduced.

Based on goals two implementations were made

- Codespace link for Task 2 first implementation program https://bug-free-space-palm-tree-vx97769j6jx266q.github.dev/

- Codespace link for Task 2 second implementation program https://bug-free-space-palm-tree-vx97769j6jx266q.github.dev/

2.4 Results

- The robot successfully navigates multiple floors.
- The agent spends time based on how dirty the room is. If the room is dirtier it takes much time to finish cleaning
- It avoids occupied rooms based on the timetable
- Skips a room after checking that it's having a makeup class.
- Randomly assignment of room status for each time the agent iterates

Conclusion

This project demonstrates the implementation of an AI-based Vacuum Cleaner Robot with increasing complexity:

1. Task 1 introduces a basic rational AI agent for two rooms.
2. Task 2 extends it to multiple rooms, floors, and real-world constraints.

The robot efficiently cleans rooms, avoids obstacles, and minimizes time, making it a rational and effective AI agent.