MALAWI UNIVERSITY OF BUSINESS AND APPLIED SCIENCE

(MUBAS)

SCHOOL OF APPLIED SCIENCES

DEPARTMENT OF COMPUTATION AND INFORMATION SYSTEMS

(ASSIGNMENT 1-YEAR 4 SEMISTER 1)

PROGRAM NAME	**: INFORMATION TECHNOLOGY**
MODULE NAME	**: ARTIFICIAL INTELLIGENCE**
MODULE CODE	**: CIT-AI-411**
SUBMITTED BY	**: BARLON DZOWA**
REG NUMBER	**: BIT/21/SS/010**
SUBMITTED TO	**: DR, AMELIA TAYLOR, PHD**
DUE DATE	**: 13TH FEBUARY, 2025**

THE VACUUM CLEANER AI ROBOT

INTRODUCTION

The Vacuum Cleaner Robot is designed AI agent in order to simulate the behaviour of an autonomous cleaning system responsible to moves between rooms, detects dirt, and cleans it accordingly. The implementation is organised into different levels, including static, random, and dynamic environments. The robot is programmed to function in a multi-room setting, considering obstacles such as occupied rooms and optimizing cleaning efficiency based on predefined constraints.

TASK 1: Basic Vacuum Cleaner Simulation using the easy method.

Is the Robot Rational? To answer this question, then the simulation of the task one was done, from the observation, A rational agent is one that takes actions that maximize its performance based on the percept sequence.

The robot continuously checks the cleanliness of each room before making a decision. In this implementation, robot was measured using the below information,

- If a room is dirty, it cleans otherwise, it moves to the next room.
- It moves sequentially between these two rooms and repeats the process for a set number of cycles.

In this context, the easy method of simulation the program of having static environment with predefined input. The robot begins with a fixed initial state where rooms have predefined cleanliness values. It moves sequentially between two rooms and cleans if necessary once the dirty is found there by making the robot to percept, cleaning the dirty and move to onother room.

TASK 2: Extending the Robot Implementation to Multiple Rooms

In this case, the rooms are structured as a dictionary, representing multiple rooms along a corridor. Then the robot moves from room to room along using the corridor.

EXPPLAINATION FROM THE CODE.

In the code, the ButterFly dictionary was crated that maps floor names to room numbers and their cleanliness. Then the VacuumCleaner class iterates through the dictionary to clean rooms. The time constraints are added to ensure that the robot accounts for the time required to travel between rooms and clean them and the aim for this is to be introduced to minimize the total cleaning time. The delay time in the code function simulates time taken to travel and clean hence the robot follows a predefined sequence thereby optimizes cleaning order.

Results

- ✓ The robot successfully cleaned the rooms in an optimized sequence.
- ✓ The cleaning cycle ensured no room was skipped and dirt is sucked and removed efficiently

TASK2 - Hard Level

The robot moves across multiple floors and uses stairs to navigate. Floors are represented as separate dictionary keys, and movement between them is simulated.

From the code, the *MUBAS_Building* dictionary defines floors with separate room lists. The robot iterates through floors, cleaning rooms before moving between levels and some rooms may be occupied (e.g., classes in session) and should not be cleaned at that time. The *room_timetable* dictionary stores room availability (1 = occupied, 0 = available), then the *is_room_available* function checks room availability before attempting to clean.

Results:

- ✓ The robot efficiently navigated three floors and cleaned all available rooms.
- ✓ It correctly skipped occupied rooms and resumed cleaning once they became available.
- ✓ Time optimization improved overall cleaning efficiency.

At this level, rooms may become dirty again after a certain period, then the room requiring continuous monitoring and cleaning as illustrated in the code.

CONCLUSION

This project successfully simulates an AI-powered vacuum cleaner robot that autonomously navigates, detects dirt, and cleans accordingly. The implementation varies in complexity, from a simple predefined environment to a dynamic multi-floor system with real-time decision-making. The rational behaviour of the robot is evident in its ability to adapt to changing conditions and optimize its cleaning strategy based on predefined constraints and real-time feedback.

NOTE: Future advances could comprise implementing machine learning algorithms to predict dirt accumulation and optimize cleaning schedules accordingly