

Assignment Specifications: Intelligent Vacuum Cleaner Agent

Objective: Develop a Python-based simulation of an intelligent agent for a vacuum cleaner navigating a 2D grid world.

Background:

In the realm of Artificial Intelligence (AI), the vacuum cleaner problem is a classic introductory problem to explain the concept of intelligent agents. Your task is to create a program that models a simple environment and a vacuum cleaner that interacts with this environment.

Requirements:

1. **Environment:**
 - Create a 2D grid world with dimensions of 5x5.
 - Randomly initialize each grid cell as either **clean** (represented by a **0**) or **dirty** (represented by a **1**).
 - Provide functionality to visualize this environment.
2. **Vacuum Cleaner Agent:**
 - Place the vacuum cleaner at an initial position, e.g., **(2, 2)**,
 - The vacuum cleaner should be able to perceive whether its current tile is dirty.
 - Based on its perception, it should decide upon an action.
3. **Actions:**
 - If the current tile is dirty, the action should be **CLEAN**, and the tile should change to **clean**.
 - If the current tile is clean, the vacuum should move. For the purpose of this assignment, the vacuum can choose a random direction (**UP, DOWN, LEFT, RIGHT**), but it should not move outside the grid boundaries.
4. **Simulation:**
 - Run a simulation where the vacuum cleaner takes 10 actions based on its perceptions of the environment.
 - After the simulation, visualize the final state of the environment.

Stretch Goals (Optional):

- **Improved Navigation:** Instead of moving randomly, implement a method for the vacuum cleaner to move towards the nearest dirty tile.
- **Obstacles:** Introduce obstacles in the environment. The vacuum cleaner should detect these obstacles and avoid them.
- **Performance Metrics:** Track the performance of your vacuum cleaner, e.g., number of tiles cleaned out of total dirty tiles in the least amount of moves.

Example Output:

Initial Environment:

```
0 1 0 1 1
0 1 1 1 0
0 0 0 0 0
0 0 0 0 1
1 1 0 1 1
```

Moving from (2, 2) to (2, 1)

Cleaning (2, 1)

Moving from (2, 1) to (3, 1)

Cleaning (3, 1)

Moving from (3, 1) to (3, 0)

Cleaning (3, 0)

Moving from (3, 0) to (4, 0)

Cleaning (4, 0)

Moving from (4, 0) to (4, 3)

Cleaning (4, 3)

After cleaning:

```
0 1 0 0 0
0 1 0 0 0
0 0 0 0 0
0 0 0 0 0
1 1 0 1 1
```

Note: This is just one example of what your output could look like. You can design your output in any way that clearly conveys the initial state, the movements and actions of the agent, and the final state.

Submission:

- Submit the source code of your program.
- Provide a brief report (1-2 pages) explaining your design decisions, challenges faced during the development, and any assumptions you made.
- If you attempt any of the stretch goals, ensure they are clearly documented in your report.
- Document your use of any AI tools in solving or coding this problem, as well as in improving your solution after you encounter any obstacles. Be sure to reflect on how the AI tools helped you in this process, and to give specific examples.

Evaluation Criteria:

Your project will be evaluated based on:

- Correctness and completion of the basic requirements.
- Code readability, structure, and comments.
- Creativity in problem-solving and the implementation of any additional features.
- Quality and depth of the submitted report.

Tips:

- Start by sketching out the design of your classes and their interactions.
- Break down the problem into smaller tasks and tackle each task one at a time.
- Test each component individually before integrating.
- Evaluate your initial results. If you see a problem like the agent making repetitive movements in a small area, explore other algorithms that might encourage it to explore more of the grid, and adjust your code accordingly.

Good luck, and enjoy the process of bringing an intelligent agent to life!