

Final Task – 2

Name: S. S. Zobaer Ahmed

ID: 22-49415-3

In this lab task, you have to design and implement an intelligent **Vacuum Cleaning Agent** using fundamental principles of Artificial Intelligence. The agent should operate in a simple two-dimensional environment (such as a grid-based world) where each cell can either be clean or dirty. The agent must be capable of navigating through the environment and cleaning the dirty cells efficiently. You must program the agent to perceive its current location's status (clean or dirty), decide its next action (move left, right, up, down, or clean), and execute the action accordingly. The agent should use a simple rule-based approach. Do the implementation using Python.

Code:

```
import numpy as np

import time

grid = np.random.choice([0, 1], size=(4, 4))

agent_pos = [0, 0]

cleaned_count = 0

def print_grid(grid, agent_pos):

    print("\n" + "-" * 20)

    for i in range(4):

        for j in range(4):

            if [i, j] == agent_pos:

                print("A", end=" ")

            else:

                print("D" if grid[i, j] == 1 else ".", end=" ")

        print()

    print("-" * 20)

# Show initial grid

print("Initial Environment:")
```

```
print_grid(grid, agent_pos)

time.sleep(1)

while not (agent_pos == [3, 3] and grid[3, 3] == 0):

    x, y = agent_pos

    if grid[x, y] == 1:

        grid[x, y] = 0

        cleaned_count += 1

        print(f"Cleaned cell at ({x}, {y})")

    if y < 3:

        agent_pos = [x, y + 1]

    elif x < 3:

        agent_pos = [x + 1, 0]

    else:

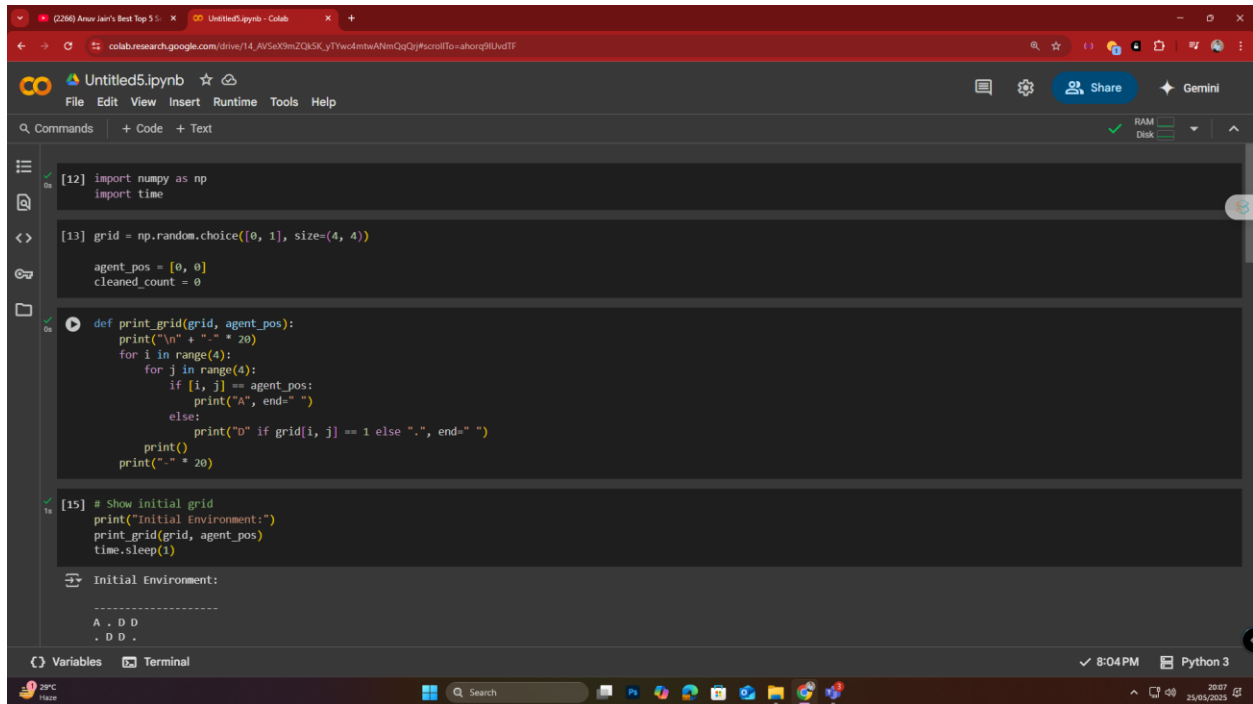
        agent_pos = [x, y]

    print_grid(grid, agent_pos)

    time.sleep(1)


print(f"Cleaning complete. Total cells cleaned: {cleaned_count}")
```

Code Screenshot:



The screenshot shows a Jupyter Notebook titled 'Untitled5.ipynb' in a web browser. The notebook contains the following code cells:

```
[12] import numpy as np
import time

[13] grid = np.random.choice([0, 1], size=(4, 4))

agent_pos = [0, 0]
cleaned_count = 0

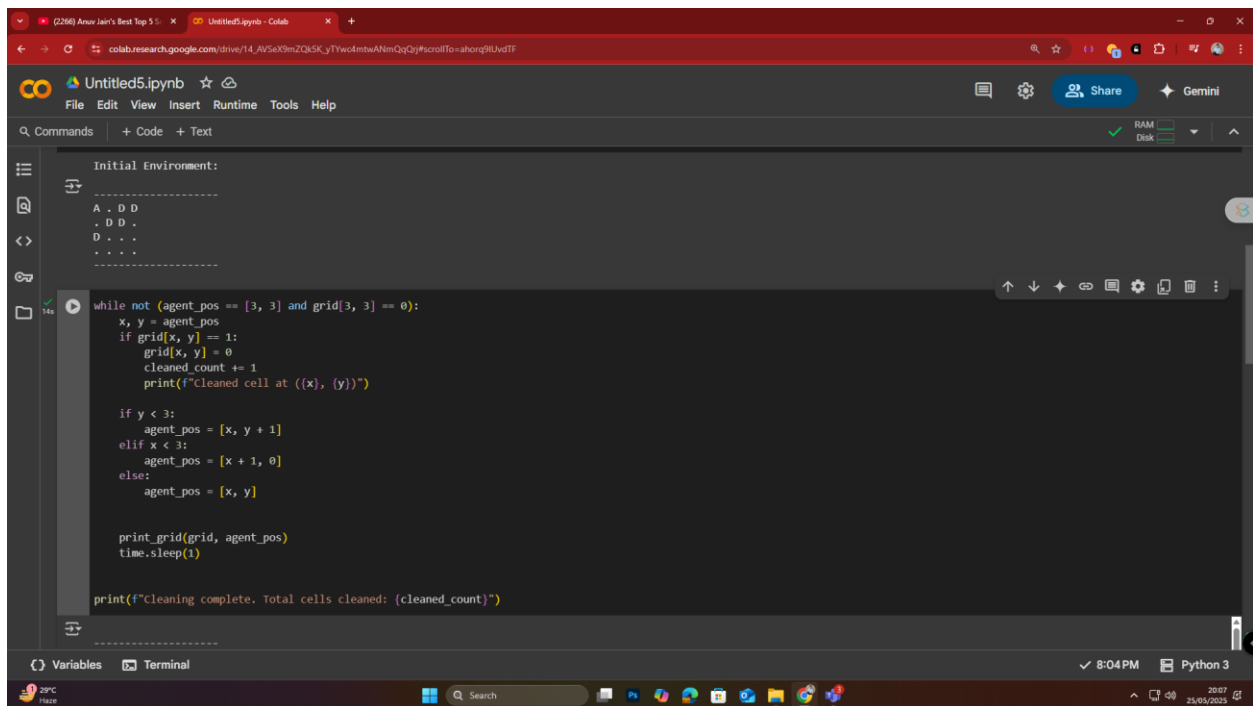
def print_grid(grid, agent_pos):
    print("\n" + "-" * 20)
    for i in range(4):
        for j in range(4):
            if [i, j] == agent_pos:
                print("A", end=" ")
            else:
                print("D" if grid[i, j] == 1 else ".", end=" ")
            print()
        print("-" * 20)

[15] # Show initial grid
print("Initial Environment:")
print_grid(grid, agent_pos)
time.sleep(1)
```

The output of the notebook shows the initial environment:

```
Initial Environment:
-----
A . D D
. D D .
D . . .
. . . .
-----
```

The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help), a toolbar with icons for commands, code, and text, and a status bar at the bottom showing the time (8:04 PM) and Python version (Python 3).



The screenshot shows the same Jupyter Notebook interface, but with the following code cell added:

```
while not (agent_pos == [3, 3] and grid[3, 3] == 0):
    x, y = agent_pos
    if grid[x, y] == 1:
        grid[x, y] = 0
        cleaned_count += 1
        print(f"Cleaned cell at ({x}, {y})")

    if y < 3:
        agent_pos = [x, y + 1]
    elif x < 3:
        agent_pos = [x + 1, y]
    else:
        agent_pos = [x, y]

    print_grid(grid, agent_pos)
    time.sleep(1)

print(f"Cleaning complete. Total cells cleaned: {cleaned_count}")
```

The output of the notebook shows the initial environment:

```
Initial Environment:
-----
A . D D
. D D .
D . . .
. . . .
-----
```

The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help), a toolbar with icons for commands, code, and text, and a status bar at the bottom showing the time (8:04 PM) and Python version (Python 3).

Output Screenshot:

The image displays two sequential screenshots of a Google Colab notebook titled 'Untitled5.ipynb'. The notebook is running on a Python 3 environment. The first screenshot shows a 3x3 grid of cells, each containing a 3x3 sub-grid of characters. The top row of cells contains the characters 'A', 'D', and 'D'. The middle row contains 'D', 'D', and 'D'. The bottom row contains 'D', 'D', and 'D'. The notebook has executed two cells, each displaying 'Cleaned cell at (0, 2)' and 'Cleaned cell at (0, 3)'. The second screenshot shows the same 3x3 grid, but now the middle row contains 'A', 'D', and 'D'. The bottom row contains 'D', 'D', and 'D'. The notebook has executed two more cells, each displaying 'Cleaned cell at (1, 1)' and 'Cleaned cell at (1, 2)'. The bottom of the notebook interface shows a prompt to 'start coding or generate with AI.'.

```
-----  
. A D .  
. D D .  
D . . .  
-----  
. A D  
. D D .  
D . . .  
-----  
Cleaned cell at (0, 2)  
  
-----  
. . A  
. D D .  
D . . .  
-----  
Cleaned cell at (0, 3)  
  
-----  
. . .  
A D D .  
D . . .  
-----  
  
-----  
. . .  
. A D .  
D . . .  
-----  
Cleaned cell at (1, 1)  
  
-----  
. . . A  
D . . .  
-----  
Cleaned cell at (1, 2)  
  
-----  
. . .  
. . . A  
D . . .  
-----  
A . . .  
-----  
  
start coding or generate with AI.
```