Lab 4 Report

Abdul Hannan Khan

Task 1: Simple Reflex Vacuum Cleaner

Explanation:

This task was about making a simple vacuum agent that decides what to do based on its environment. The code checks whether the current spot is “Clean” or “Dirty.” If it’s dirty, the agent chooses “Suck,” and if it’s clean, it chooses “Move.” This solves the problem by letting the agent react correctly to each situation without any complex logic.

Code:

Task1_SimpleVacuumAgent.py > ...

```python
class VacuumAgent:
    def action(self, sense):
        place, position = sense

        if position == "Dirty":
            return "Suck"
        elif position == "Clean":
            return "Move"

simpleVacuum = VacuumAgent()

print(simpleVacuum.action(("A", "Clean")))
print(simpleVacuum.action(("B", "Dirty")))
print(simpleVacuum.action(("C", "Dirty")))
print(simpleVacuum.action(("D", "Clean")))
```

Output:

```
[Running] python -u "d:\Coding Languages\Python Language\Lab Tasks\Task1_SimpleVacuumAgent.py"
Move
Suck
Suck
Move

[Done] exited with code=0 in 0.399 seconds
```

Task 2: Traffic Light Reflex Agent

Explanation:

This task was about creating an agent that reacts to different traffic light colors. The code checks whether the light is Red, Yellow, or Green. Based on the color, the agent returns “Stop,” “Slow,” or “Go.” This solves the problem by making the agent follow basic traffic rules automatically.

Code:

Task2_TrafficLightAgent.py > ...

```python
class TrafficAgent:
    def action(self, color):
        traffic, light = color

        if light == "Red":
            return "Stop"
        elif light == "Yellow":
            return "Slow"
        elif light == "Green":
            return "Go"

trafficLight = TrafficAgent()

print(trafficLight.action(("A", "Green")))
print(trafficLight.action(("B", "Yellow")))
print(trafficLight.action(("C", "Red")))
```

Output:

```
[Running] python -u "d:\Coding Languages\Python Language\Lab Tasks\Task2_TrafficLightAgent.py"
Go
Slow
Stop

[Done] exited with code=0 in 0.324 seconds
```

Task 3: Model-Based Vacuum Cleaner Agent

Explanation:

This task was about creating a model-based vacuum agent that remembers the status of both rooms. The code updates the room's condition every time the agent senses it, and then decides what to do. If the room is dirty, it returns "Suck," and if both rooms are clean, it does "NoAction." Otherwise, it chooses "Move." This solves the problem by letting the agent use memory to make smarter decisions.

Code:

Task3_ModelVacuumAgent.py > ModelVaccumAgent

```python
class ModelVaccumAgent:
    def __init__(self):
        self.room_status = {"A": "Dirty", "B": "Dirty"}

    def action(self, sense):
        place, position = sense

        self.room_status[place] = position

        if position == "Dirty":
            return "Suck"

        if self.room_status["A"] == "Clean" and self.room_status["B"] == "Clean":
            return "NoAction"

        return "Move"

modelVacuum = ModelVaccumAgent()

print(modelVacuum.action(("A", "Dirty")))
print(modelVacuum.action(("A", "Clean")))
print(modelVacuum.action(("B", "Dirty")))
print(modelVacuum.action(("B", "Clean")))
```

Output:

```
[Running] python -u "d:\Coding Languages\Python Language\Lab Tasks\Task3_ModelVacuumAgent.py"
Suck
Move
Suck
NoAction

[Done] exited with code=0 in 0.239 seconds
```

Task 4: Model-Based Warehouse Robot Agent

Explanation:

This task was about creating a warehouse agent that remembers which shelves it has already picked items from. The code checks whether a shelf has a package and whether the agent has picked from that shelf before. If the shelf has a package and it's the first time, the agent chooses "Pick," otherwise it says "Skip." If there's no package, it returns "Move." This solves the problem by letting the agent avoid picking the same item twice.

Code:

Task4_WarehouseAgent.py > ModelWarehouseAgent > action

```python
class ModelWarehouseAgent:
    def __init__(self):
        self.picked_items = {}

    def action(self, sense):
        place, hasPackage = sense

        if hasPackage == "Yes":
            if place not in self.picked_items:
                self.picked_items[place] = True
                return "Pick"
            else:
                return "Skip"
        else:
            return "Move"

warehouse = ModelWarehouseAgent()
print(warehouse.action(("Shelf1", "Yes")))
print(warehouse.action(("Shelf1", "Yes")))
print(warehouse.action(("Shelf2", "No")))
print(warehouse.action(("Shelf2", "Yes")))
print(warehouse.action(("Shelf2", "Yes")))
```

Output:

```
[Running] python -u "d:\Coding Languages\Python Language\Lab Tasks\Task4_WarehouseAgent.py"
Pick
Skip
Move
Pick
Skip

[Done] exited with code=0 in 0.224 seconds
```