

Date: 1/10/24

Program Title: Implement Vacuum Cleaner Agent

Algorithm -

1/10/24

IMPLEMENT VACUUM CLEANER AGENT

ALGORITHM

1. Start with the agent at an initial location (A or B)
2. For each percept
 - ① If the location is dirty, perform the action Suck
 - ② If the location is clean, move to the other location
 - (a) If the agent is at A, move right to B
 - (b) If the agent is at B, move left to A
3. Repeat for each new percept until no dirty squares remain
4. If there is no further action needed, do NOOP (no operation)

OUTPUT:

Percept: ['A', 'Clean'], Action: Right
Percept: ['A', 'Clean', 'Dirty'], Action: Suck
Percept: ['B', 'Clean'], Action: Left
Percept: ['B', 'Dirty'], Action: Suck
Percept: ['A', 'Clean'], Action: Right
Percept: ['A', 'Clean'], Action: Right
Percept Sequence: [['A', 'Clean'], ['A', 'Dirty'], ['B', 'Clean'], ['B', 'Dirty'], ['A', 'Clean'], ['A', 'Clean']]
Action Sequence: ['Right', 'Suck', 'Left', 'Suck', 'Right', 'Right']

Code:

```

def vacuum_cleaner_agent(percept):
    location, status = percept

    if status == 'Dirty':
        return 'Suck'
    elif location == 'A':
        return 'Right'
    elif location == 'B':
        return 'Left'
    else:
        return 'NoOp'
percepts = [['A', 'Clean'], ['A', 'Dirty'], ['B', 'Clean'], ['B', 'Dirty'], ['A', 'Clean'], ['A', 'Clean']]
actions = []
for percept in percepts:
    action = vacuum_cleaner_agent(percept)
    actions.append(action)
    print(f"Percept: {percept}, Action: {action}")
print("\nPercept Sequence:", percepts)
print("Action Sequence:", actions)

```

```

def vacuum_cleaner_agent(percept):
    location, status = percept

    if status == 'Dirty':
        return 'Suck'
    elif location == 'A':
        return 'Right'
    elif location == 'B':
        return 'Left'
    else:
        return 'NoOp'
percepts = [['A', 'Clean'], ['A', 'Dirty'], ['B', 'Clean'], ['B', 'Dirty'], ['A', 'Clean'], ['A', 'Clean']]
actions = []
for percept in percepts:
    action = vacuum_cleaner_agent(percept)
    actions.append(action)
    print(f"Percept: {percept}, Action: {action}")
print("\nPercept Sequence:", percepts)

```

Snapshot of the output

Percept Sequence: [['A', 'Clean'], ['A', 'Dirty'], ['B', 'Clean'], ['B', 'Dirty'], ['A', 'Clean'], ['A', 'Clean']]
Action Sequence: ['Right', 'Suck', 'Left', 'Suck', 'Right', 'Right']

OUTPUT:

Percept: ['A', 'Clean'], Action: Right
Percept: ['A', 'Clean', 'Dirty'], Action: Stop
Percept: ['B', 'Clean'], Action: left
Percept: ['B', 'Dirty'], Action: Stop
Percept: ['A', 'Clean'], Action: Right
Percept: ['A', 'Clean'], Action: Right
Percept Sequence: [['A', 'Clean'], ['A', 'Dirty'],
['B', 'Clean'], ['B', 'Dirty'], ['A', 'Clean'],
['A', 'Clean']]
Action Sequence: ['Right', 'Stop', 'left', 'Stop',
'Right', 'Right']

STATE SPACE TREE:

