Date / /
Page

01-10-24

Algorithm for Vaccum cleaner

→ Firstly considering two rooms for the cleaning
→ check and make sure that the vaccum cleaner should clean the two rooms and returns to the initial state again after the completion of two rooms.
→ Conider location of the room and Status of the room

0 → represents clean
1 → represents dirty

Vaccum cleaner can move left, Right, up and down because it will be working in 2d grid

```
def is dirty ():
return self.room [self.position] = 1

def is clean ():

if room.self.is dirty ():
    self.room [self.position] = 0
    self.cleaned = room + = 1

def move ():
self.position = 1 - self.position
```

Date / /
Page

- 10 - 24

cleaner

rooms for

t the vacuum
he two rooms
al state again
o room,
e room and

an
ty

e left,
because
2d grid

position) = 1

):
] = 0
= 1

ition

def run(steps):

for step in range(steps):
clean()
move()
rooms = [1, 0]

Percept Sequence

Check: RoomA, Dirty
Action: Clean RoomA & move

Check: RoomB, Dirty
Action: Clean RoomB & move

I :→ (Room 1, Dirty)
II :→ (Room 2, clean)
III → (Room 1, clean)
IV → (Room 2, clean)
V → (Room 1, clean)
VI → (Room 2, dirty)

Code:

class Vacuumcleaner:
def __init__(self, rooms, start_position):

self.rooms = rooms
self.position = start_position
self.cleaned_rooms = 0

Date __/__/__
Page _____

```
self.percept_sequence = []

def is_dirty(self):
    return self.room[self.position] == 1

def clean(self):
    if self.is_dirty():
        print(f"Cleaning room {self.position + 1}")

def move(self):
    if self.position < len(self.room) - 1:
        self.position += 1
    else:
        self.position = 0
    print(f"Moved to room {self.position + 1}")

def perceive(self):
    room_state = "Dirty" if self.is_dirty() else "Clean"
    percept = (f"Room {self.position + 1}", room_state)
    self.percept_sequence.append(percept)
    print(f"Perception: {percept}")

def run(self, steps):
    for step in range(steps):
        print(f"Step {step + 1}:")
        self.perceive()
        self.clean()
        self.move()
```

```
print(f"Room status : {self.rooms}\n")
print(f"Total cleared rooms: {self.
    cleared_rooms}")
print("Percept sequence :", self.percept_
  sequence)
rooms = [1, 0, 1, 1]
vacuum = VacuumCleaner(rooms,
  start_position = 0)
vacuum.run(steps = 8)
```

Output

Step 1:
Perception : ('Room 1', 'Dirty')
Cleaning Room 1
Move to room 2
Room status : [0, 0, 1, 1]

Step 2:
Perception : ('Room 2', 'Clean')
Moved to Room 3
Room status : [0, 0, 1, 1]

Step 3:
Perception : ('Room 3', 'Dirty')
Cleaning Room 3
Moved to room 4
Room status : [0, 0, 0, 1]

Step 4:
Perception : ('Room 4', 'Dirty')
Cleaning room 4
Moved to room 1

Date __/__/__
Page

Step 5:
Perception: ('Room 1', 'Clean')
Moved to room 2
Room Status: {0, 0, 0, 0}

Step 6:
Perception: ('Room 2', 'Clean')
Moved to room 3
Room Status: {0, 0, 0, 0}

Step 7:
Perception: ('Room 3', 'Clean')
Moved to room 4
Room Status: {0, 0, 0, 0}

Step 8:
Perception: ('Room 4', 'Clean')
Moved to room 1
Room Status: {0, 0, 0, 0}

Total cleaned room: 3
Percept sequence:
{(Room 1, Dirty)
(Room 2, Clean)
(Room 3, Dirty)
(Room 4, Dirty)
(Room 1, Clean)
(Room 2, Clean)
(Room 3, Clean)
(Room 4, Clean)}

27/10/24