

LAB-2

Vacuum Cleaner Algorithm

Step-1

Create 2 rooms using classes

Class room:

```
def __init__(self, a):
```

```
    self.state = a
```

```
def suck(self):
```

```
    self.state = "clean"
```

Step-2

Instantiate the class and take user input

```
a = str(input("Room A state"))
```

```
b = str(input("Room B state"))
```

```
room_list = []
```

```
room_list.append(room(a))
```

```
room_list.append(room(b))
```

Step-3

Perception sequence

```
for i in room_list:
```

```
    if (i.state == "dirty"):
```

```
        i.suck()
```

Proceed

Code -

class VacuumCrawlerAgent:

def __init__(self):

Initial State

self.state = {

'RoomA': 'Dirty',

'RoomB': 'Dirty',

'VacuumPosition': 'RoomA'

}

~~self.state = {'RoomA': 'Dirty', 'RoomB': 'Dirty', 'VacuumPosition': 'RoomA'}~~

def display_environment(self):

print('Room A: { self.state[\'RoomA\'] } | Room B: { self.state[\'RoomB\'] } | Vacuum: { self.state[\'VacuumPosition\'] }')

def clean(self):

current_position = self.state[\'VacuumPosition\']

if self.state[current_position] == 'Dirty':

self.state[current_position] = 'Clean':

else

print("{current_position} is already clean")

def move(self):

current_position = self.state[\'VacuumPosition\']

if current_position == 'RoomA':

self.state[\'VacuumPosition\'] = 'RoomB'

print("Moving to Room B")

'A moat' - ('mole' / 'mural') state . fls

("A moat at ground") fls

: (fls) state . fls

'mole' - ('mole' / 'mural') state . fls

: (fls) state . fls

: (fls) state . fls

(fls) state . fls

(fls) state . fls

: (fls) state . fls

(fls) state . fls

('A moat at ground') fls

(fls) state . fls

(fls) state . fls

-TUTUO

fls : A moat

fls : A moat

A moat : mural

A moat : mural

fls : A moat

fls : A moat

fls : A moat

Use:

```
self.state['VacuumPosition'] = 'Room A'
```

```
print("Moving to Room A")
```

```
def is_clean(self):
```

```
    return self.state['Room A'] == 'Clean' and self.state['Room B'] == 'Clean'
```

```
def run(self):
```

```
    while not self.is_clean():
```

```
        self.display_environment()
```

```
        self.clean()
```

```
        if not self.is_clean():
```

```
            self.move()
```

```
print("Both rooms are clean now")
```

```
self.display_environment()
```

```
vacuum_agent = VacuumCleanerAgent()
```

```
vacuum_agent.run()
```

OUTPUT-

Room A: Dirty

Room B: Dirty

Vacuum: Room A

Cleaning Room A.

Room A: Clean

Cleaning Room B

Both rooms are clean now.