

Daren Liu
00001471643

Lab Assignment #2: Vacuum Cleaner Agent - Report

Explanation of the defined functions:

main.py:

Main.py first imports the roomba class from vacuum.py. It then prompts the user to enter 3 values, either 0 or 1s. These values are then stored in a list called states, which will be used to describe the state of the tiles and what position the vacuum is in. obj is then declared and calls the constructor for the class roomba. Obj then runs the member function check(), and prints out the list of actions that the obj takes with obj.todo, as well as how many steps it took with obj.score.

vacuum.py:

In vacuum.py, the class roomba is declared, with lists states and todo expected to be passed in. The class then initializes variables in its own scope with self and sets them equal to states and todo. There's also another variable called score that is stored in the class. There is then a member function check() which has a while loop to have the vacuum cleaning as long as one tile is dirty. There are then if statements that account for each possible input except for [0,0,0] and [0,0,1], since this means that both tiles are clean and there are no actions needed. The code would keep executing an if statement and looping again until all tiles are clean.

Explanation of the test case:

When the user is prompted to enter three inputs in main.py, the first value indicates whether the left tile is clean or dirty, with 0 indicating that it's clean and 1 indicating that it's dirty. Similarly, the second value indicates whether the right tile is clean or dirty, with the same values and meaning. The third value indicates whether the vacuum is in the left or right tile, with 0 being the left tile and right being the right tile.

[0,0,0] and [0,0,1]:

No action needed since both tiles are clean.

Actions of the vacuum: []

Performance of the vacuum: 0

[1,1,0]:

The vacuum would need to clean, move to the right tile and then clean

Actions of the vacuum: ['suck', 'move right', 'suck']

Performance of the vacuum: 3

[1,1,1]:

The vacuum would need to clean, move to the left tile and then clean

Actions of the vacuum: ['suck', 'move left', 'suck']

Performance of the vacuum: 3

[0,1,0]:

The vacuum would need to move to the right tile, and then clean

Actions of the vacuum: ['move right', 'suck']

Performance of the vacuum: 2

[1,0,1]:

The vacuum would need to move to the left tile and then clean

Actions of the vacuum: ['move left', 'suck']

Performance of the vacuum: 2

[0,1,1]:

The vacuum would just need to clean the current tile

Actions of the vacuum: ['suck']

Performance of the vacuum: 1

[1,0,0]:

The vacuum would just need to clean the current tile

Actions of the vacuum: ['suck']

Performance of the vacuum: 1

Appendix:

main.py

```
from vacuum import roomba

def main():
    val1 = input("Enter 0 or 1: ")
    val2 = input("Enter 0 or 1: ")
    val3 = input("Enter 0 or 1: ")
    states = [int(val1), int(val2), int(val3)]

    obj = roomba(states, [])
    obj.check()
    print("Actions of the vacuum: ", obj.todo)
    print("Performance of the vacuum: ", obj.score)

if __name__ == '__main__':
    main()
```

vacuum.py

```
class roomba:
    def __init__(self, states, todo):
        self.states = states
        self.todo = todo
        self.score = 0
```

```

def check(self):
    while self.states[0] == 1 or self.states[1] == 1:
        if self.states[0] == 1 and self.states[1] == 1 and self.states[2] == 0:
#both dirty and roomba is left 110
            self.states[0] = 0
            self.todo.append("suck")
            self.states[2] = 1
            self.todo.append("move right")
            self.score += 2
        elif self.states[0] == 1 and self.states[1] == 1 and self.states[2] == 1:
#both dirty and roomba is right 111
            self.states[1] = 0
            self.todo.append("suck")
            self.states[2] = 0
            self.todo.append("move left")
            self.score += 2
        elif self.states[0] == 0 and self.states[1] == 1 and self.states[2] == 1:
#right dirty and roomba is right 011
            self.states[1] = 0
            self.todo.append("suck")
            self.score += 1
        elif self.states[0] == 1 and self.states[1] == 0 and self.states[2] == 0:
#left dirty and roomba is left 100
            self.states[0] = 0
            self.todo.append("suck")
            self.score += 1
        elif self.states[0] == 0 and self.states[1] == 1 and self.states[2] == 0:
#right dirty, and roomba is left 010
            self.states[2] = 1
            self.todo.append("move right")
            self.states[1] = 0
            self.todo.append("suck")
            self.score += 2
        elif self.states[0] == 1 and self.states[1] == 0 and self.states[2] == 1:
#left dirty, and roomba is right 101
            self.states[2] = 0
            self.todo.append("move left")
            self.states[0] = 0
            self.todo.append("suck")

```

```
self.score += 2
```

```
#0 is clean, 1 is dirty
```

```
#For positions, 0 is left, 1 is right
```

```
#All possible states
```

```
#0, 0, 0 doesnt matter
```

```
#0, 0, 1 doesnt matter
```

```
#0, 1, 0 done
```

```
#1, 0, 0 done
```

```
#0, 1, 1 done
```

```
#1, 1, 0 done
```

```
#1, 0, 1 done
```

```
#1, 1, 1 done
```