

Department of Computer Science
Course: Artificial Intelligence
Quiz-01

Student Name: Elnazar Ulanbek uulu

Write a program for Multi-Agent Environment. You can modify the program written for the Vacuum Cleaner (Program file is available in the Moodle).

```
from random import choice, randint, shuffle

class Vacuum:

    __slots__ = (
        '__model',
        '__score'
    )

    def __init__(self, model):
        self.__model = model
        self.__score = 0

    @property
    def model(self):
        return self.__model

    @property
    def score(self):
        return f"{self.__model}'s score {self.__score}!"

    def suck(self, location: str, status: str, *args, **kwargs):
        print(f'location {location} is {status}, {self.__model} is going to clean it!')
        print(f'location {location} has been cleaned!')
        self.__score += 1

    def move(self, location: str, direction: str, status: str, *args, **kwargs):
        print(f'location {location} is {status}, {self.__model} has moved to the {direction}!')

    def __repr__(self):
        return f'Vacuum({self.__model})'

class Location:

    __slots__ = ('__location',
                 '__status')

    def __init__(self, location=None, status=None):
        if all([location is None, status is None]):
            self.__location, self.__status = self.__call__()
        else:
            self.__location = location
            self.__status = status

    def __call__(self, *args, **kwargs):
```

```

        location = choice(('A', 'B'))
        status = choice(('Dirty', 'Clean'))
        return location, status

    @property
    def location(self):
        return self.__location

    @property
    def status(self):
        return self.__status

    def __repr__(self):
        return f'Location({self.__location}, {self.__status})'

def reflex_vacuum_agent(vacuum, location: str, status: str) -> object:
    if status.lower() == 'dirty':
        return vacuum.suck(location=location, status=status.lower())
    if location.lower() == 'a':
        return vacuum.move(location=location, direction='right',
status=status.lower())
    return vacuum.move(location=location, direction='left',
status=status.lower())

def reflex_multi_vacuum_agent(rooms, *vacuums):
    while rooms:
        shuffle(rooms)
        room = rooms.pop()
        vacuum = choice(vacuums)
        print(f"{'-' * 64}\n{room=}\n{vacuum=}\n{vacuum.model}'s logger:")
        reflex_vacuum_agent(
            vacuum=vacuum,
            location=room.location,
            status=room.status,
        )
    print(f'\nResult:')
    for vacuum in vacuums:
        print(vacuum.score)

def main():
    rooms = [Location() for _ in range(randint(3, 5))]
    print(f'Randomly generated {rooms=}')
    vacuum_1 = Vacuum('Mark-1')
    vacuum_2 = Vacuum('T900')
    # vacuum_3 = Vacuum('Ilkhomzhon') # he will never know it! ahahah
    reflex_multi_vacuum_agent(rooms, vacuum_1, vacuum_2)

if __name__ == '__main__':
    main()

```

Everything works randomly!

RESULT:

```
D:\Programming\UCA\venv\Scripts\python.exe D:\Programming\UCA\main.py
Randomly generated rooms=[Location(A, Dirty), Location(B, Dirty), Location(A, Dirty),
Location(B, Dirty), Location(A, Dirty)]
-----
room=Location(A, Dirty)
vacuum=Vacuum(T900)
T900's logger:
location A is dirty, T900 is going to clean it!
location A has been cleaned!
-----
room=Location(B, Dirty)
vacuum=Vacuum(T900)
T900's logger:
location B is dirty, T900 is going to clean it!
location B has been cleaned!
-----
room=Location(A, Dirty)
vacuum=Vacuum(T900)
T900's logger:
location A is dirty, T900 is going to clean it!
location A has been cleaned!
-----
room=Location(B, Dirty)
vacuum=Vacuum(T900)
T900's logger:
location B is dirty, T900 is going to clean it!
location B has been cleaned!
-----
room=Location(A, Dirty)
vacuum=Vacuum(Mark-1)
Mark-1's logger:
location A is dirty, Mark-1 is going to clean it!
location A has been cleaned!

Result:
Mark-1's score 1!
T900's score 4!

Process finished with exit code 0
```