

### Final Year B. Tech (EE)

**Trimester: VII Subject: AIML**

### Name: Shreerang Mhatre

### Roll No: 52

**Class: Final Year B. Tech (EE)**

**Batch: A3**

A3

# Experiment No: 03

**Name of the Experiment**: Vacuum Cleaner World Agent using Python

### Performed on: 11/09/2023



**Marks Teacher’s Signature withdate**

### Submitted on: 11/09/2023



**Aim:** To develope a simple reflex agent program in Python for the vacuum-cleaner world problem.

**Prerequisite:** Knowledge of Agents.

### Objective:

To create a Vacuum Cleaner World Agent using Python Programming.

### Components and Equipment required: Python

**Exp. 3- 1**

**Theory:**

**Vacuum cleaner problem** is a well-known search problem for an agent which works on

ArtificialIntelligence. In this problem, our vacuum cleaner is our agent. It is a goal based agent, and the goal of this agent, which is the vacuum cleaner, is to clean up the whole area. So, in the classical vacuum cleaner problem, we have two rooms and one vacuum cleaner. There is dirt in both the rooms and it is to be cleaned. The vacuum cleaner is present in any one of these rooms. So, we have to reach a state in which both the rooms are clean and are dust free.

This program defines the States, Goal State, Goal Test, Actions, Transition Model, and Path Cost. For each possible initial state, the program returns a sequence of actions that leads to the goal state, along with the path cost. Generates two test cases.

1.

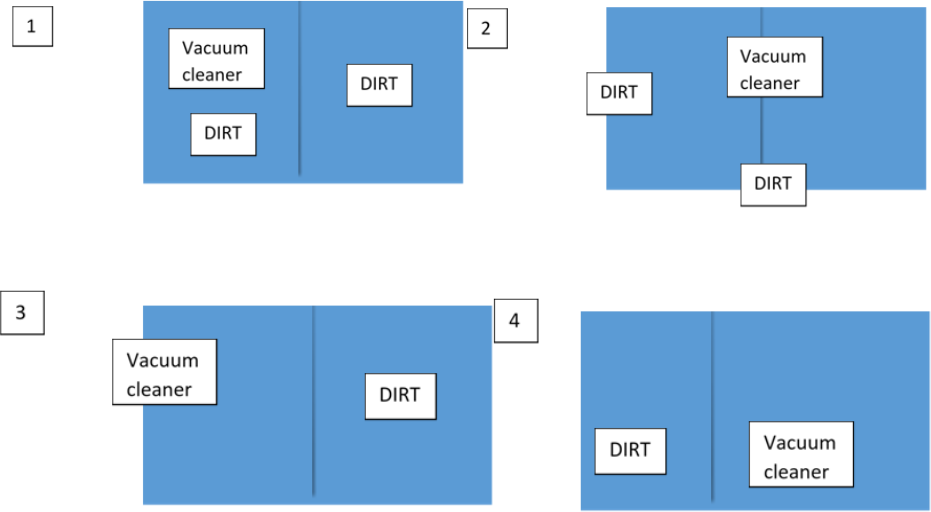
EnterLOCATIONA/BincaptialletterswhereAandBarethetwoadjace ntrooms respectively.

1. EnterStatusO/1accordinglywhere0meansCLEANand1meansDIRTY.

3.

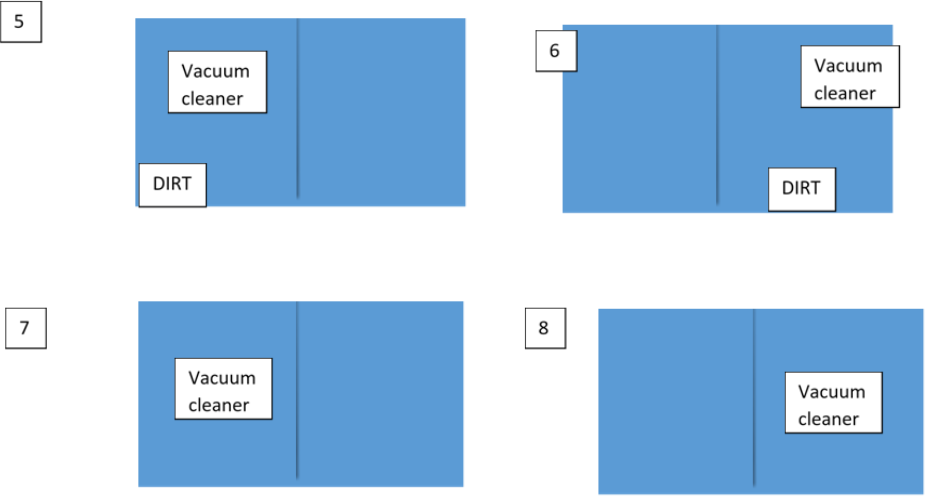
VacuumCleanersensesthestatusoftheotherroombeforeperforminga nyaction, also known as Environmentsensing.

So,thereareeightpossiblestatespossibleinour**vacuumcleanerproblem**.Thesecanbewe ll illustratedwiththehelpofthefollowingdiagrams:



**Exp. 3- 2**





Here, states 1 and 2 are our initial states and state 7 and state 8 are our final states (goal states). This means that, initially, both the rooms are full of dirt and the **vacuum cleaner** can reside in any room. And to reach the final goal state, both the rooms should be clean and the **vacuum**

**cleaner** again can reside in any of the two rooms.

The **vacuum cleaner** can perform the following functions: move left, move right, move forward, move backward and to suck dust. But as there are only two rooms in our problem, the vacuum

cleaner performs only the following functions here: move left, move right and suck.

Here the performance of our agent (vacuum cleaner) depends upon many factors such as time taken in cleaning, the path followed in cleaning, the number of moves the agent takes intotal, etc. But we consider two main factors for estimating the performance of the agent. Theyare:

* 1. **Search Cost:** How long the agent takes to come up with thesolution.
  2. **Path cost:** How expensive each action in the solutionare.

By considering the above factors, the agent can also be classifies as a utility based agent.



Procedure:

* Create an Environment and Vacuum Cleaner Agent
* Select random location and random status for Vacuum Cleaner Agent
* Verify Initial Location and Environment Status.
* Achieve the goal state.

## Programme Code

import random

class Environment(object):

definit(self): self.locationcondition={'A'

: '1' , 'B' :'1'} #RANDOM CONDITION

self.locationcondition['A']=random.randint(0,1)

self.locationcondition['B']=random.randint(0,1)

class Sreflexagent(Environment): definit(self,Environment):

#print(Environment.locationcondition) #place vacum at random location vacuumlocation=random.randint(0, 1) #if vacuum at A

if vacuumlocation==0:

print("vacuum is randomly placed at locationn A") #and if location A is dirty

if Environment.locationcondition['A']==1: print("Location A is dirty")

#suck the dirt and mark it clean Environment.locationcondition['A']=0 print("Location A has been cleaned") print("moving to location B") vacuumlocation=1

else:

print("Location A is clean") #move to B

print("moving to location B") vacuumlocation=1



if

Environment.locationcondition['B']==1

: #suck the dirt and mark it clean

Environment.locationcondition['B']=0 print("Location B has been cleaned")

else:

print("Location B is clean") elif vacuumlocation==1:

print("vacuum is randomly placed at locationn B") #and if location B is dirty

if Environment.locationcondition['B']==1: print("Location B is dirty")

#suck the dirt and mark it clean Environment.locationcondition['B']=0; print("Location B has been cleaned") print("moving to location A") vacuumlocation=0

else:

print("Location B is Clean") #Move to A

print("moving to location A") vacuumlocation=0

if vacuumlocation==0:

if Environment.locationcondition['A']==1: print("Location A is dirty")

#suck the dirt and mark it clean Environment.locationcondition['A']=0; print("Location A has been cleaned")

else:

print("Location A is Clean")

#DONE CLENING

#print(Environment.locationcondition) theEnvironment=Environment() thevacuum=Sreflexagent(theEnvironment)

**Exp. 3- 5**

[**WWW.MITWPU.COM**](http://www.mitwpu.com/)



**Conclusion:**

A vacuum cleaner world agent is developed for two locations considering random status using Python Programming.

# Post Lab Questions:

## What are the different types of agents? Explain the pseudocode for utility based agent.

1. Explain different terms related to agent.

## what are the different types of environment?

**Exp. 3- 6**

[**WWW.MITWPU.COM**](http://WWW.MITWPU.COM)

