***Artificial Intelligence***

***CSL 411***

***Lab Journal 3***

****

**Student Name**

**Enrolment No.**

**Class and Section**

**Department of Computer Science**

**BAHRIA UNIVERSITY, ISLAMABAD**

**Lab # 3: Rational Agents**

**Objectives:**

To implement Simple Reflex & Model Based Agent in Vacuum World.

**Tools Used:**

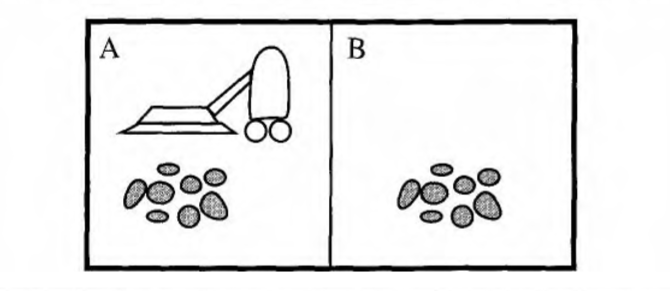
Python IDLE 3.4/Python IDLE 3.6

**Submission Date:**

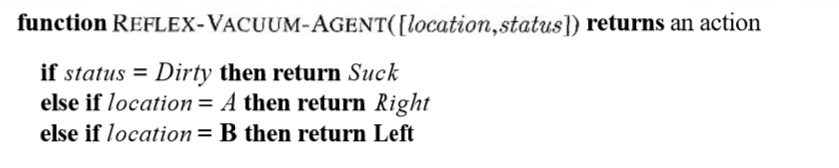
**Evaluation: Signatures of Lab Engineer:**

**Task # 1:**

Consider the vacuum world shown in the figure below:



This particular world has just two locations: squares A and B. The vacuum agent perceives which square it is in and1 whether there is dirt in the square. It can choose to move left, move right, suck up the dirt, or do nothing. One very simple agent function is the following: if the current square is dirty, then suck, otherwise move to the other square. A simple program for the agent function of vacuum-world is shown below:



Your task is to implement the above vacuum world and its agent program for a simple reflex agent. Also, suggest a performance measure and evaluate your program based on that performance measure. Modify your program accordingly.

**Procedure/Program:**

import random

status = ['clean', 'dirty']

location = ['A', 'B']

action = ['suck', 'nothing']

def Reflex\_Vacuum\_Agent(locat,status,action):

    loc = locat[random.randint(0, 1)]

    stts = status

    while stts != ['clean', 'clean']:

        stts = status

        state\_A = stts[random.randint(0, 1)]

        state\_B = stts[random.randint(0, 1)]

        if state\_A != 'clean':

            act = action[0]

            print(f"\nA is {state\_A} Do {act} ")

            state\_A = stts[0]

            loc = locat[1]

            act = action[1]

            print(f"\nA is {state\_A} Do {act} ")

            print(f"\nNow location is {loc}")

            if state\_B == 'clean':

                act = action[1]

                print(f"\nB is {state\_B} Do {act} ")

                print('Both rooms are clean now')

                stts=[state\_A,state\_B]

            else:

                continue

        elif state\_A == 'clean':

            act=action[1]

            print(f"\nA is {state\_A} Do {act} ")

            loc = locat[1]

            print(f"\nNow location is {loc}")

            if state\_B == 'clean':

                act = action[1]

                print(f"\nB is {state\_B} Do {act} ")

                print('Both rooms are clean now')

                stts=[state\_A,state\_B]

            else:

                continue

        elif state\_B != 'clean':

            act = action[0]

            print(f"\nB is {state\_B} Do {act} ")

            state\_B = stts[1]

            loc = locat[0]

            print(f"\nB is {state\_B} Do {act} ")

            print(f"\nNow location is {loc}")

            if state\_A == 'clean':

                act = action[1]

                print(f"\nA is {state\_A} Do {act} ")

                print('Both rooms are clean now')

                stts=[state\_A,state\_B]

            else:

                continue

        elif state\_B == 'clean':

            act = action[1]

            print(f"\nB is {state\_B} Do {act} ")

            loc = locat[0]

            print(f"\nNow location is {loc}")

            if state\_A == 'clean':

                act = action[1]

                print(f"\nA is {state\_A} Do {act} ")

                print('Both rooms are clean now')

                stts=[state\_A,state\_B]

            else:

                continue

Reflex\_Vacuum\_Agent(location,status,action)

**Result/Output:**

Graphical user interface, text, application, email

Description automatically generated

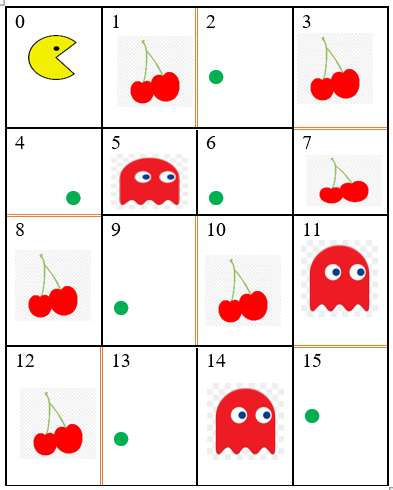
**Analysis/Conclusion:**

**Task # 2:**

Given a simple pacman game in figure below that consisting of 4\*4 grid. The starting point of pacman is cell 0 and its goal is to consume/eat maximum food pallets, while considering following given limitations.

* Pacman can move up, down, left right (keeping in view walls).
* Pacman can eat power pallets, i.e., cherries to keep ghost scared, i.e., if pacman enters the ghost cell its is not destroyed.
* Pacman keeps moving until all the power pallets are consumed.

You need to devise a **model/goal-based agent** for the above given problem.



**Procedure/Program:**

**Result/Output:**

**Analysis/Conclusion:**

**Task # 3:**

Develop a medical diagnosis system, designed as a **simple reflex agent** that diagnose the disease on the basis of provided symptoms and test reports. Symptoms and test reports should be taken from the user as percepts and agent has to display the diagnosed disease as its action. Also suggest that how can you convert this agent into **model-based agent**, what changes from implementation perspective can be done to convert it into model based.

**Acute appendicitis:**

Symptoms: Fever, Pain in Abdomen especially ILIAC FOSSA, vomiting,

Test: Blood CP with ESR… TLC (Total leucocyte count) will be high, DLC (Differential leucocyte count) Neutrophils will be high , ESR high

Treatment: Surgery

**Pneumonia:**

Symptoms: Fever, Cough (with sputum), Pain in chest

Blood CP with ESR… TLC (Total leucocyte count) will be high, DLC (Differential leucocyte count) Neutrophils will be high , ESR high

X-ray chest: pneumonic patch (sometimes)

Treatment: Antibiotics

**Acute Tonsilitis:**

Symptoms: Fever, Cough

Test: Examine throat: (Red enlarged tonsils, pus in tonsils)

Treatment: anti-allergic, paracetamol. If not gone, add antibiotics orally. If not gone, add antibiotics IV

**Procedure/Program:**

**Result/Output:**

**Analysis/Conclusion:**