**EXPERIMENT 02: Simple Reflex Agent Vacuum Cleaner**

CLASS: BE CMPN A ROLL NO. : 19

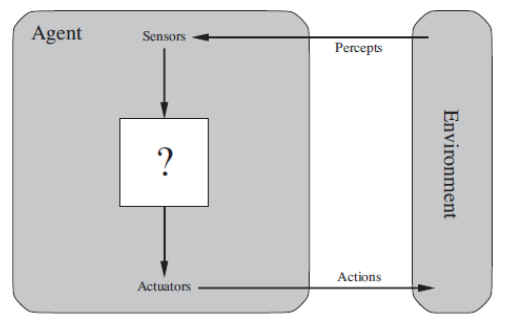
NAME: REBECCA DIAS PID: 182027

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

**Theory:**

**Agents and Environment**

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators. Agents can be hardware or software.  Software agents are called softbots.



**Types of Agents**

**1. Human Agent:**

**Sensors:** eyes, ears, skin

**Actuators:** hands, legs, vocaltract and so on

**2. Robotic Agent:**

**Sensors:** camera, infrared finders

**Actuators:** motors

**3. Software Agent:**

**Sensors:** keystrokes, mouse clicks, files and network packets

**Actuators:** screen display, writing files, sending network packets and so on

**Percept**

* It is the complete set of input that an intelligent agent is perceiving at any given moment.
* The current percept or the sequence of percept can influence the actions of the agent

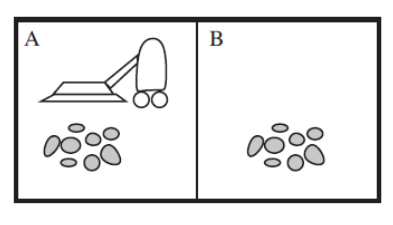
**Percept Sequence**

It is the complete history of everything the agent has perceived. An agent’s choice of action at any given instant can depend on the entire percept sequence observed to date.

**AGENT FUNCTION**

* An agent’s behavior is described by the **agent function** that maps any given percept sequence to an action.
* Tabulating the agent function is an external characterization and this would be a very large infinite table.

**VACUUM CLEANER WORLD PROBLEM**

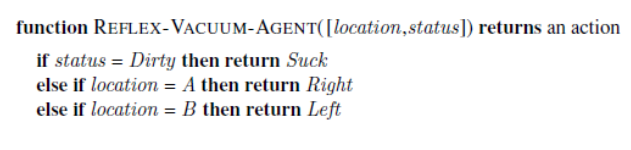


* This particular world has just two locations: squares A and B.
* The vacuum agent perceives which square it is in and whether there is dirt in the square.
* It can choose to move left, move right, suck up the dirt, or do nothing.
* The simple agent function is :
* If the current square is dirty, then suck
* Else, move to the other square.

**Agent Program**

The agent function for an artificial agent will be implemented by an agent program. The agent function is an abstract mathematical description; the agent program is a concrete implementation, running within some physical system.

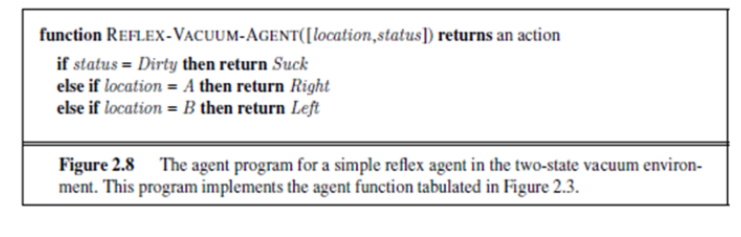
For Vacuum Cleaner



**Simple Reflex Agents**

* Simplest
* Acts on current percept
* Ignores history

**Agent Program for Vacuum Cleaner**



* Vacuum agent program is very small because
  + Ignoring the percept history (cuts down no. of possibilities )
  + when the current square is dirty, the action does not depend on the location.
* It can be implemented using condition–action rule
* E.g.
* **if** car-in-front-is-braking

**then** initiate-braking.

* Agent program: Build a general-purpose interpreter for condition–action rules and then create rule sets for specific task environments.

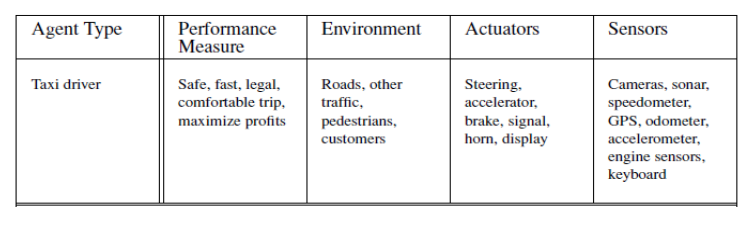
**Task Environment**

Task Environments are essentially the “problems” to which rational agents are  “solutions.”

**Specifying the task Environment : PEAS Description**

The task environment of an AI agent is specified by performance measure, environment, and the agent’s actuators and sensors. This is acronymically given as the PEAS (Performance, Environment, Actuators, and Sensors) description. In designing an agent, the first step must always be to specify the task environment as fully as possible.

**Example: Automated Taxi Driver**



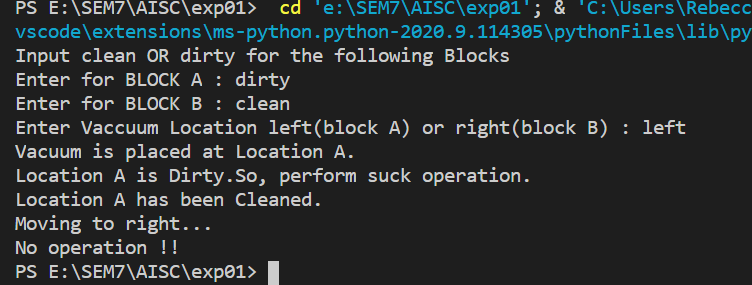
**Experiment Exercise** :

1. Develop a simple reflex agent program for the vacuum-cleaner world problem. The vacuum cleaner world is defined as follows: There are 2 states (left, right), initially a vacuum cleaner could be in any one state. There are 3 operations: **suck** if dirty, **move to right** if left state is clean and **move to left** if right state is clean). If both the states are clean then exit (**Nooperation**).

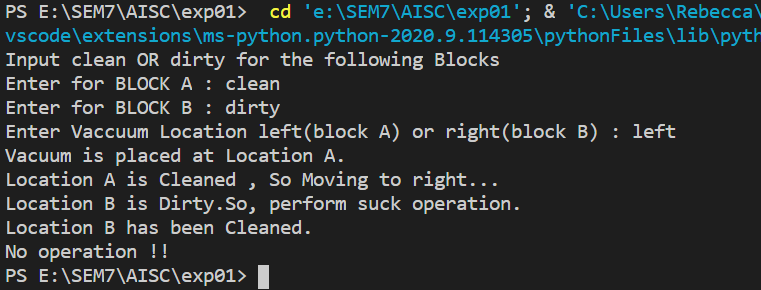
Code:

|  |
| --- |
| class Env(object):      def \_\_init\_\_(self):          print("Input clean OR dirty for the following Blocks")          self.loc = {'A': 'clean', 'B': 'clean'}          self.loc['A'] =  input("Enter for BLOCK A : ")          self.loc['B'] =  input("Enter for BLOCK B : ")  class SimpleReflexVacuumAgent(Env):      def \_\_init\_\_(self, Env):            vacuumLocation =  input("Enter Vaccuum Location left(block A) or right(block B) : ")          if(Env.loc['A'] == "clean" and Env.loc['B'] == "clean"):              print("No operation !!")              return          if vacuumLocation == "left":              print("Vacuum is placed at Location A.")                if Env.loc['A'] == "dirty":                  print("Location A is Dirty.So, perform suck operation.")                    Env.loc['A'] = "clean"                  print("Location A has been Cleaned.")                    print("Moving to right...")                    if Env.loc['B'] == "dirty":                      print("Location B is Dirty.")                        Env.loc['B'] = "clean"                      print("Location B has been Cleaned.")              else:                    print("Location A is Cleaned , So Moving to right...")                    if Env.loc['B'] == "dirty":                      print("Location B is Dirty.So, perform suck operation.")                        Env.loc['B'] = "clean"                      print("Location B has been Cleaned.")                  else:                      print("Location B is Clean")          elif vacuumLocation == "right":              print("Vacuum  placed at Location B.")                if Env.loc['B'] == "dirty":                  print("Location B is Dirty.")                    Env.loc['B'] = "clean"                  print("Location B has been Cleaned.")                    print("Moving to left...")                    if Env.loc['A'] == "dirty":                      print("Location A is Dirty.")                        Env.loc['A'] = "clean"                      print("Location A has been Cleaned.")              else:                    print("Location B is Cleaned , So Moving to left...")                    if Env.loc['A'] == "dirty":                      print("Location A is Dirty.")                        Env.loc['A'] = "clean"                      print("Location A has been Cleaned.")                  else:                      print("Location A is cleaned")          else:              print("Please enter a valid input!")              return    Environment = Env()  VacuumCleanerAgent = SimpleReflexVacuumAgent(Environment) |

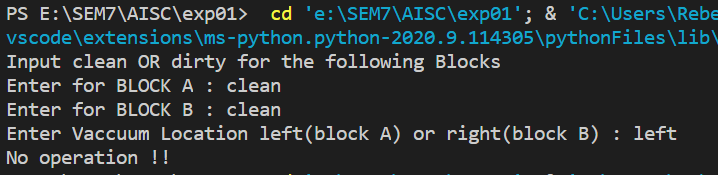
Output:

//Block A is dirty vacuum on left

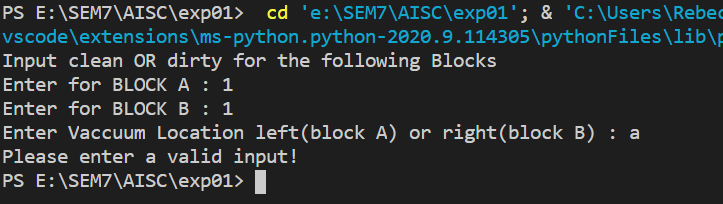
//Block B is dirty vacuum on left



//both clean



// Invalid input



**Post Experiment Exercise:**

1. Identify a real robotic vacuum cleaner in the market and define the PEAS description for it.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AGENT | PERFORMANCE | ENVIRONMENT | ACTUATOR | SENSOR |
| Mila Grow iMap Robotic Floor Cleaner | Cleanness, Efficiency, Battery life, Security | Room, Table, Wood floor, Carpet, Various obstacles | Wheels, Brushes, Vacuum Extractor | Wheels, Brushes, Vacuum Extractor |

**Conclusion:** In this experiment, an automated vacuum cleaner was designed to clean 2 block in an environment. The cleaner can detect and clean dirt on the blocks. It can also move between blocks as it sees fit. This agent was based on a problem to clean 2 states using a simple reflex agent. The agent worked perfectly and desired output was obtained.