**COEN 166 Artificial Intelligence**

**Lab Assignment #2: Vacuum World - Sample Report**

Name: Kevin Hom ID: 00001268661

**Explanation of the defined functions:**

Def \_\_init\_\_ this function takes in the state and stores the location and state of the agent in its member variables

Def agent\_function takes in three parameters itself, the location, and the status of the location. This function maps the action associated with the given percept. If the space is dirty then clean, else move either left or right depending if the current space it is in is dirty.

Def space\_is\_clean function checks whether both spaces are clean and returns true if it is and false if it is not. This function is used to repeatedly check if the state space is clean so the program knows when to terminate

**Explanation of the test case:**

For the test case is used to test whether the program is able to perform the correct actions given its current input.

Test case 1: [“Clean”, “Clean”, 1],[]

Output: [],0

Test case 2: [“Clean”,”Dirty”,0],[]

Output: [“Right”, “Suck”], 2

Test case 3: [“Dirty”, “Dirty”, 1],[]

Output: [“Suck”, “Left”, “Suck”], 3

With the given test cases, the user can calculate the result by hand or in their head and this is used to verify whether or not the program is performing its computations correctly and outputs the correct output.

**Appendix:**

class ReflexAgent:

def \_\_init\_\_(self,state):

self.state = state

self.initial\_location = state[2]

def agent\_function(self,location, status):

if status == "Dirty":

return "Suck"

elif location == 0 and status == "Clean":

return "Right"

elif location == 1 and status == "Clean":

return "Left"

def space\_is\_clean(self):

if self.state[0] == "Clean" and self.state[1] == "Clean":

return "true"

else:

return "false"

state = ["Clean", "Clean",1] #Hardcode the state here

actions = []

num\_actions = 0

Agent = ReflexAgent(state)

while Agent.space\_is\_clean() != "true":

current\_action = Agent.agent\_function(Agent.initial\_location, state[Agent.initial\_location])

num\_actions += 1

actions.append(current\_action)

if current\_action == "Right":

Agent.initial\_location = 1

elif current\_action == "Left":

Agent.initial\_location = 0

else:

Agent.state[Agent.initial\_location] = "Clean"

print(actions, num\_actions)

#print('Space is cleaned!')