**Virutal Initio Programming: WS28 & Ex28 Sample Answers**

**WS28**

**General Troubleshooting:** Watch out for complaints about apostraphes if students are cutting and pasting between the worksheet and IDLE.

**Question 1:** A dictionary is printed which will look something like:

{'distance': 630.0, 'obstacle\_right': 0, 'obstacle\_left': 0, 'line\_left': 0, 'line\_right': 0}

Though values may vary.

**Question 2:** This will depend upon the set up. In the beliefbase above the value is 630.

**Question 3:** The dictionary now contains 'test': 1

**Question 4:** 'name': 'initio'has been added.

**Question 5:** There is no ‘name’ key in the dictionary.

**Question 6:** It returns True because distance is a key in the dictionary.

**Question 7:** It prints out the value of the distance sensor stored in the belief base.

**Question 8:** It prints a list containing the string ‘a\_goal’

**Question 9:** It prints an empty list. check\_goals removed ‘a\_goal’ from the goalbase because ‘a\_goal’ was now an agent belief.

**Question 10:** The simulated Initio moves forward and then stops – just like typing initio.forward(10) etc. in earlier exercises. This is because the robot is a field in the agent object.

**Ex28**

**Exercise 1:**

import bdi.initioagent as cognitive

agent = cognitive.InitioAgent()

agent.init()

agent.getPercepts()

while not (agent.beliefbase['line\_left']):

 if (agent.beliefbase['obstacle\_left']):

 agent.robot.spinLeft(10)

 elif (agent.beliefbase['obstacle\_right']):

 agent.robot.spinLeft(10)

 else:

 agent.robot.forward(10)

 agent.getPercepts()

agent.robot.stop()

**Exercise 2:**

import bdi.initioagent as cognitive

agent = cognitive.InitioAgent()

agent.init()

while True:

 agent.getPercepts()

 distance = agent.beliefbase['distance'];

 if (distance < 50):

 agent.add\_belief('obstacle')

 elif ('obstacle' in agent.beliefbase):

 agent.remove\_belief('obstacle')

 if ('obstacle' in agent.beliefbase):

 agent.robot.spinLeft(10)

 else:

 agent.robot.forward(10)

**Exercise 3:**

import bdi.initioagent as cognitive

agent = cognitive.InitioAgent()

agent.init()

agent.getPercepts()

agent.add\_goal('obstacle\_left')

while ('obstacle\_left' in agent.goalbase):

 agent.robot.forward(10)

 agent.getPercepts()

 agent.check\_goals()

agent.robot.stop()

**Exercise 4:**

import bdi.initioagent as cognitive

agent = cognitive.InitioAgent()

agent.init()

agent.getPercepts()

while not (agent.beliefbase['obstacle\_left']):

 if (agent.beliefbase['line\_left'] and not (agent.beliefbase['line\_right'])):

 agent.robot.forward(10)

 elif (agent.beliefbase['line\_right']):

 agent.robot.spinRight(10)

 else:

 agent.robot.spinLeft(10)

 agent.getPercepts()

agent.robot.stop()

**Exercise 5:**

import bdi.initioagent as cognitive

agent = cognitive.InitioAgent()

agent.init()

agent.getPercepts()

def edge\_following():

 while ('edge\_square' in agent.goalbase):

 if (agent.beliefbase['line\_left'] and not (agent.beliefbase['line\_right'])):

 agent.robot.forward(10)

 elif (agent.beliefbase['line\_right']):

 agent.robot.spinRight(10)

 else:

 agent.robot.spinLeft(10)

 agent.getPercepts()

 if (agent.beliefbase['obstacle\_left']):

 agent.drop\_goal('edge\_square')

def find\_edge():

 agent.add\_goal('line\_left')

 while ('line\_left' in agent.goalbase):

 agent.robot.forward(10)

 agent.getPercepts()

 agent.check\_goals()

agent.add\_goal('edge\_square')

find\_edge()

edge\_following()

agent.robot.stop()



 University of Liverpool, 2020

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).