**Virtual Initio Programming: WS16 Sample Answers and Trouble Shooting**

**WS16**

**Exercise 1:**  Note that the students won’t see much movement here since it doesn’t turn very far before it loses the obstacle. They might want to alter the distances so they seem more activity.

import simclient.simrobot as initio

import time

def shake\_head():

initio.setServo(1, 20)

time.sleep(5)

initio.setServo(1, -20)

time.sleep(5)

initio.setServo(1, 0)

initio.init()

while (not initio.getDistance() < 50):

 initio.forward(10)

initio.stop()

shake\_head()

while (initio.getDistance() < 50):

 initio.spinLeft(10)

 time.sleep(5)

initio.stop()

shake\_head()

while (not initio.getDistance() < 50):

 initio.forward(10)

initio.stop()

shake\_head()

A cleverer answer to Exercise 1 uses a second function as follows:

import simclient.simrobot as initio

import time

def shake\_head():

initio.setServo(1, 20)

time.sleep(5)

initio.setServo(1, -20)

time.sleep(5)

initio.setServo(1, 0)

def while\_no\_obstacle():

 while (not initio.getDistance() < 50):

 initio.forward(10)

 initio.stop();

initio.init()

while\_no\_obstacle()

shake\_head()

while (initio.getDistance() < 50):

 initio.spinLeft(10)

 time.sleep(5)

initio.stop()

shake\_head()

while\_no\_obstacle()

shake\_head()

**Question 1&2:** The turn(side) function turns left or right depending upon its argument. The following program turns the robot right for 10 seconds.

**Exercise 2:**

def turn\_obstacle(side):

 if (side == 'right'):

 initio.spinLeft(10)

 while(initio.irRight()):

 continue

 else:

 initio.spinRight(10)

 while(initio.irLeft()):

 continue

 initio.stop()

initio.init()

turn\_obstacle('left')

**Question 3:** returns the value of the obstacle sensor on the left or right or depending upon its argument.

**Exercise 3:**

import simclient.simrobot as initio

import time

def obstacle(side):

 if (side == 'left'):

 return initio.irLeft()

 else:

 return initio.irRight()

initio.init()

print(obstacle('left'))

print(obstacle('left'))

**Exercise 4:**

def opposite(side):

 if (side == 'left'):

 return 'right'

 else:

 return 'left'

**Exercise 5:**

import simclient.simrobot as initio

import time

def obstacle(side):

 if (side == 'left'):

 return initio.irLeft()

 else:

 return initio.irRight()

def turn(side):

 if (side == 'left'):

 initio.spinLeft(10)

 else:

 initio.spinRight(10)

def turn\_until(side):

 turn(side)

 while(obstacle(opposite(side))):

 continue

 initio.stop();

def opposite(side):

 if (side == 'left'):

 return 'right'

 else:

 return 'left'

initio.init()

turn\_until('left')

**Ex16**

**Exercise 1:**

import simclient.simrobot as initio

import time

def when\_obstacle\_close():

 while (not initio.getDistance() < 15):

 continue

 time.sleep(3)

initio.init()

when\_obstacle\_close()

initio.forward(10)

when\_obstacle\_close()

initio.stop()

**Exercise 2:**

import simclient.simrobot as initio

import time

def forward\_for(t):

 initio.forward(10)

 time.sleep(t)

 initio.stop()

initio.init()

t1 = input("Enter a time in seconds")

forward\_for(int(t1))

**Exercise 3:**

import simclient.simrobot as initio

import time

def average\_distance():

 count = 0

 total = 0

 while (count < 10):

 total = total + initio.getDistance()

 time.sleep(1)

 count = count + 1

 return total/10

initio.init()

initio.forward(10)

average = average\_distance()

initio.stop()

print(str(average))

**Exercise 4:**

def follow\_line():

 while True:

 while (initio.irLeftLine()):

 initio.spinLeft(10)

 while (initio.irRightLine()):

 initio.spinRight(10)

 while (not initio.irLeftLine() and not initio.irRightLine()):

 initio.forward(10)

**Exercise 5:**

import simclient.simrobot as initio

import time

def avoid\_obstacle():

 while (not initio.getDistance() < 15):

 while (initio.getDistance () < 50 and initio.getDistance() > 15):

 initio.spinLeft(10)

 while (initio.getDistance() > 50):

 initio.forward(10)

def follow\_line():

 while (not initio.getDistance() < 15):

 while (initio.irLeftLine() and not initio.getDistance() < 15):

 initio.spinLeft(10)

 while (initio.irRightLine() and not initio.getDistance() < 15):

 initio.spinRight(10)

 while (not initio.irLeftLine() and not initio.irRightLine() and not initio.getDistance() < 15):

 initio.forward(10)

initio.init()

while True:

 todo = input("Obstacle or Line or Stop? O/L/S")

 if (todo == "O"):

 avoid\_obstacle()

 elif (todo == "L"):

 follow\_line()

 else:

 break

 initio.stop()

**Exercise 6:** Note the use of a sleep command to give the servo time to turn. In the simulator this program may never end.

import simclient.simrobot as initio

import time

def furthest():

 initio.setServo(1, 20)

 time.sleep(1)

 left = initio.getDistance()

 initio.setServo(1, -20)

 time.sleep(1)

 right = initio.getDistance()

 initio.setServo(1, 0)

 print(str(left))

 print(str(right))

 if (left > right):

 return 'left'

 elif (left < right):

 return 'right'

 else:

 return 'neither'

def turn(side):

 if (side == 'left'):

 initio.spinLeft(10)

 elif (side == 'right'):

 initio.spinRight(10)

 else:

 initio.stop()

initio.init()

while (not furthest() == 'neither'):

 turn(furthest())

initio.stop()



 University of Liverpool, 2019

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