**Virtual Pi2Go Programming: WS19 Sample Answers and Trouble Shooting**

**Question:** The program moves forward for 3 seconds, then backwards for 3 seconds, then turns left for 3 seconds, then turns right for 3 seconds and then stops.

**Sample Answer Exercise 1:**

import simclient.simrobot as pi2go

import time

actions = ['forward', 'backward', 'left', 'right', 'stop']

pi2go.init()

for action in actions:

 if (action == 'forward'):

 pi2go.forward(10)

 time.sleep(3)

 elif (action == 'backward'):

 pi2go.reverse(10)

 time.sleep(3)

 elif (action == 'left'):

 pi2go.spinLeft(10)

 time.sleep(3)

 elif (action == 'right'):

 pi2go.spinRight(10)

 time.sleep(3)

 else:

 pi2go.stop()

**Sample Answer Exercise 2:**

import simclient.simrobot as pi2go

import time

actions = ['forward', 3, 'backward', 2, 'left', 1, 'right', 5, 'stop']

pi2go.init()

act = 1

for el in actions:

 if (act == 1):

 if (el == 'forward'):

 pi2go.forward(10)

 elif (el == 'backward'):

 pi2go.reverse(10)

 elif (el == 'left'):

 pi2go.spinLeft(10)

 elif (el == 'right'):

 pi2go.spinRight(10)

 else:

 pi2go.stop()

 act = 0

 else:

 time.sleep(el)

 act = 1

**Sample Answer Exercise 3:**

import simclient.simrobot as pi2go

import time

pi2go.init()

pi2go.forward(10)

readings = []

while (len(readings) < 30):

 reading = pi2go.getDistance()

 readings.append(reading)

 time.sleep(3)

pi2go.stop()

for reading in readings:

 print(str(reading))

**Sample Answer Exercise 4:**

import simclient.simrobot as pi2go

import time

actions = ['forward', 'backward', 'left', 'right', 'stop']

pi2go.init()

pi2go.forward(10)

while (len(actions) > 0):

 for action in actions:

 if (action == 'forward'):

 pi2go.forward(10)

 time.sleep(3)

 elif (action == 'backward'):

 pi2go.reverse(10)

 time.sleep(3)

 elif (action == 'left'):

 pi2go.spinLeft(10)

 time.sleep(3)

 elif (action == 'right'):

 pi2go.spinRight(10)

 time.sleep(3)

 else:

 pi2go.stop()

 time.sleep(3)

 if (pi2go.getDistance() < 50):

 actions.remove(action)

 print("removing " + action)

pi2go.stop()

**Sample Answer Exercise 5:**

import simclient.simrobot as pi2go

import time

actions = ['forward', 'backward', 'left', 'right', 'stop']

permitted = ['backward', 'left', 'stop']

pi2go.init()

for action in actions:

 if (action in permitted):

 if (action == 'forward'):

 pi2go.forward(10)

 time.sleep(3)

 elif (action == 'backward'):

 pi2go.reverse(10)

 time.sleep(3)

 elif (action == 'left'):

 pi2go.spinLeft(10)

 time.sleep(3)

 elif (action == 'right'):

 pi2go.spinRight(10)

 time.sleep(3)

 else:

 pi2go.stop()

 else:

 print(action + " is not permitted")

**Sample Answer Exercise 6:**

import simclient.simrobot as pi2go

import time

actions = ['forward', 'backward', 'left', 'right', 'stop']

data = []

pi2go.init()

for action in actions:

 before = pi2go.getDistance()

 if (action == 'forward'):

 pi2go.forward(10)

 time.sleep(3)

 elif (action == 'backward'):

 pi2go.reverse(10)

 time.sleep(3)

 elif (action == 'left'):

 pi2go.spinLeft(10)

 time.sleep(3)

 elif (action == 'right'):

 pi2go.spinRight(10)

 time.sleep(3)

 else:

 pi2go.stop()

 time.sleep(3)

 after = pi2go.getDistance();

 data.append([action, before, after])

print(data)



 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/).