**Virtual Pi2Go Programming: WS12 Sample Answers**

**Sample Answer: 1** The program will take two readings from the ultrasonic (distance) sensor at a 10 second interval. If the first reading is less than the second reading it will print out “Object is moving away”. If the first reading is greater than the second reading it will print out “Object is moving closer”.

**Sample Answer 2:** There are three things to test. Firstly I can put a block in front of the robot, not move it, and run the program. It should print out “Object is not moving”. Secondly I can move the block after the program has started running. It should print out either “Object is moving away” or “Object is moving closer” depending upon whether I move the block closer or further away. I should test both these options.

**Potential Issues/Trouble Shooting:**

* Cut and paste of program from the work sheet may create syntax errors (particularly to do with the use of “ and indentation inside if statements)
* The simulator initialisation doesn’t complete until after initialisation complete is printed. At this point the first reading is immediately taken. Students who move the block to eagerly may find they get “Object is not moving” - they need to wait for the initialisation complete message before moving the block.
* Students have 10 seconds to move the block after initialisation is complete. This ought to be plenty of time but students do need to be aware of it.

**Sample Answer Exercise 1:** Note the use of time.sleep(10) and pi2go.stop()are not necessary to successfully complete the exercise, but they do make a nicer program.

import simclient.simrobot as pi2go, time

pi2go.init()

reading1 = pi2go.getDistance()

time.sleep(10)

reading2 = pi2go.getDistance()

if (reading1 < reading2):

 pi2go.forward(10)

 time.sleep(10)

 pi2go.stop()

**Sample Answer Exercise 2:** The elif isn’t necessary but does showcase the use of !=

import simclient.simrobot as pi2go, time

pi2go.init()

reading1 = pi2go.getDistance()

time.sleep(10)

reading2 = pi2go.getDistance()

if (reading1 == reading2):

 print("Object Stopped!")

elif (reading2 != reading1):

 print("Object Moving!")

**Sample Answer Exercise 3:** Note I’ve reduced the sleep time in order to make the robot a bit more responsive.

Students may find it useful to use print statements to see the values of reading1 and reading2 in order to debug their programs.

import simclient.simrobot as pi2go, time

pi2go.init()

while (not pi2go.getSwitch()):

 reading1 = pi2go.getDistance()

 time.sleep(3)

 reading2 = pi2go.getDistance()

 if (reading1 < reading2):

 pi2go.forward(10)

 else:

 pi2go.stop()

pi2go.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/).