**Virtual Initio Programming: WS1-4 Sample Answers and Troubleshooting**

**WS1**

**Answers:** The answers to the question on WS1 are mostly descriptive.

* World items can be moved around by clicking and dragging.
* Lines have to be moved/deleted by clicking in the centre of the whole shape.
* Blocks can not be moved over the robot.
* Black squares and lines can move under the robot.

**Potential Problems:** Moving between the Object Window and the Simulator Window is not entirely intuitive and may involve more clicks than users anticipate.

**WS2**

**Sample Answers:** The questions here are all descriptive and should be easy to answer provided the student is watching the robot and doesn’t enter the commands too rapidly one after the other.

**Potential Problems:**

* Confusion may be caused by students failing to press Return after entering a command.

**WS3**

**Sample Answers:** The questions here are all descriptive and should be easy to answer once the student has correctly understood what to type.

**Potential Problems:**

* Confusion may be caused by students failing to press Return after entering a command.
* Confusion may be caused by the use of argument for *speed* etc. in the commands. This may need to be explained to students.
* Weaker students may not like the open nature of the final question and perhaps will need direction to try out a specific set of commands and see what they do.

WS4

**Sample Answers:** The questions here are all descriptive again and the difficulties may lie in understanding how to change the values returned by the various sensors. Particularly with manipulating the objects from the Object Window.

**Exercises 4 Worlds**

**Potential Problems:**

* Confusion about starting and stopping the simulator and where to type initio.cleanup() and initio.init() particularly if they have been using IDLE to start the simulator as well as to control the robot.

cleanup() and init() exist to cleaning stop and start the connection to the simulator from the exercise window in IDLE (the equivalent functions on the real robot establish connections to the sensors and motors). If cleanup() isn’t called then errors about sockets already being in used (Address already in use) will be raised. If init isn’t called then the robot will not respond to commands.

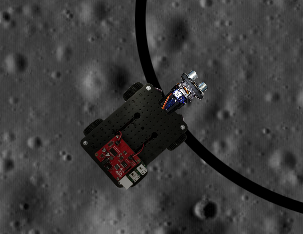
**Sample Answer Exercise 1:** For the first question the answers should be 90.0 (for the distance sensor) and 0 for both the IR sensors.

For the second question: When the robot hits a block it is stopped and judders slightly as it continues to hit the block. This often makes it move sideways and sometimes means it eventually gets around the block.

**Sample Answer Exercise 2:** When the sensor is facing directly to the wall, setServo(1, 0), the distance is 30 as the servo rotates this gradually increases as the distance between the wall and the sensor increases (to 40, 50, 60) until it suddenly increases a lot in the negative direction where the sensor is pointing at the gap before reducing as the sensor detects the wall on the right of the robot. On the left the values increase until the sensor is pointing to the corner and then decrease as it points to the edge of the world. (NB. The exact numbers may vary depending on where the children placed the robot, but the important thing is that they understand how the sensor values are changing depending upon where it is pointing).

**Sample Answer Exercise 3:** Both sensors should return 0.

**Potential Problems:** It can be very fiddly to get the robot correctly positioned over the black line, particularly if the user isn’t entirely sure where the sensor are. The picture below shows one correct placement.





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/).