**Virtual Pi2Go 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.
* The Light source can be placed and then its ray moved to point towards 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.
* Watch out for the difference between lower case and upper case letter Ls in setAllLEDs
* 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 pi2go.cleanup() and pi2go.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 1:** The two commands both give the same answer. The exact value depends upon the position of the light source. They give the same answer because they both check the same sensor FL means (front left) and the 0 supplied as an input to getLight(0)tells the system the number of the sensor to use and the front left sensor is number 0.

There are three other paired light sensor commands: getLight(1) and getLightFR(), getLight(2) and getLightBR(), and getLight(3) and getLightBL().

**Sample Answer Exercise 3:** Both sensors should return 1.

**Potential Problems:**  There is a bug which means the simulator things the two line detection sensors for the Pi2Go are a short distance in front of the robot. This should be fixed in the next version of the software.



 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/).