**Pi2Go Programming: Obstacle Avoidance**

**AIM:** After completing this worksheet you should be able to integrate your Python programming knowledge to create obstacle avoidance behaviour in your Pi2Go robot.

**You Need:** To complete this worksheet you need to have a Pi2Go that is connected to a keyboard, mouse and monitor (see WS1), to understand how to start and stop IDLE from the Linux Command Line (see WS2), and to be able to use files to store Programs (WS5). You also need to know the commands to operate the Pi2Go motors, LEDs and sensors (WS3 & WS4). You should be able to use If statements (WS7) and while loops (WS8) in Python programs.

**Challenge:** Implement obstacle avoidance behaviour on a Pi2Go robot.

**Designing the program:** Before you can implement a behaviour, you should try to define what that behaviour involves. In this challenge we will define obstacle avoidance behaviour to mean: “The Pi2Go should move forwards while there is no obstacle and should turn (either right or left) when there is an obstacle”.

We can illustrate this with a flow chart

Is there an obstacle?

Turn

Forward

Yes

No

**Question 1:** What sensor and motion commands do you expect to use for this challenge?

You can go ahead and attempt to write the program now – or you can follow the suggested steps below.

**Step 1:** Write a python program containing a while loop, which will make your Pi2Go move forward while there is no obstacle and then stop when an obstacle appears. Test your program works.

**Step 2:** Extend your program so that you have a Python program that will move forward while there is no obstacle, then will turn while there is an obstacle, and then stops. Test your program works.

**Step 3:** Now you want to extend the program so instead of going around one while loop (while there is no obstacle) and then a second (while there is an obstacle) and then stopping, it instead repeats going around the two while loops. You can do this with a third while loop that contains the other two and which starts with while True:

**Question 2:** Write out how you are going to test that your program works.

**Question 3:** Execute your program. Does it work? YES/NO

**Extending the Program:** While you have successfully implemented obstacle avoidance behaviour, the behaviour of your program isn’t as nice as you might like. For instance, you have to use Ctrl-C to stop the program running and then stop the Pi2Go at the Python command line.

**Exercise 1:** Extend the program so you can stop and start it easily using the switch.

**Remember:** When you have finished working with the robot type pi2go.cleanup() at the command, exit IDLE, and select Shutdown from the Raspberry Pi menu item. Once the robot has shut down, switch it off.



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/).