**Virtual Initio Programming: Functions Exercises**

**AIM:** This exercise sheet provides additional programs using functions.

**Exercise 1:** Create a function when\_obstacle\_close() which waits until and obstacle is closer than 15cm from the distance sensor before continuing – and sleeps for 3 seconds to give time for the obstacle to be moved away. Illustrate its use in a program which starts the robot moving once and obstacle appears briefly in front of it and stops the robot when obstacle appears again.

You can imagine this allowing you to stop and start the program by waving a hand in front of the distance sensor. It is easy to “wave” an obstacle in front of the robot sensors using the mouse and blocks from the edit window.

**Exercise 2:** Create a function forward\_for(t) where t is *time in seconds*. Which moves the robot forward for t seconds. Illustrate its use in a program which prompts the user to enter a time in seconds and then moves the robot forward for that number of seconds.

**Exercise 3:** Create a function average\_distance() which returns the average value from the ultrasonic sensor taken over ten readings taken 1 second apart. Illustrate its use in a program that drives the robot forwards, takes ten distance readings, stops and prints out the average distance.

**Exercise 4:** Create a function follow\_line() that moves forward if neither line sensor detects anything and turns left if the left line sensor detects something and right if the right line sensor detects something. Test this out in **line\_following.xml** world by placing the robot so that its line sensors are either side of the black line.

**Exercise 5:** Create a function avoid\_obstacle() that implements obstacle avoidance behaviour and a function follow\_line(). Illustrate their use in a program that prompts the use to select whether the robot should either avoid obstacles or follow a line or exit the program. If an obstacle is moved closer than 10cm then the program should prompt for a new instruction.

**Exercise 6:** Write a function furthest() which returns either ‘left’ or ‘right’ depending upon which side an obstacle is further away (you will need to use the servo to pan the distance sensor to decide this). Illustrate its use with a program that will turn towards the away from obstacles using the turn(side) function from the worksheet.



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/).