

# CS 1301 – Mini Homework: Avoiding Obstacles

## This is a Pair programming assignment!

---

You are expected to work with the person you have been paired with in class. Your pair may collaborate with other students in this class. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for other pairs. Collaboration at a reasonable level will not result in substantially similar code. For pair programming assignments, you and your partner should turn in identical assignments.

### **Avoid Walls Code**

There are various kinds of sensors on the robot: light sensor (detect the how bright the light is), proximity sensor (see whether there is anything around the robot), stall sensor, the camera, and the battery voltage. We are going to use the robot's proximity sensors for this assignment. Be sure to read all of the information below. It explains many functions that are vital to the successful completion of this assignment.

### **Mission:**

Your robot will be randomly placed in an arena of size 5 x 3 (Units: 11 in). You need to write a function called **avoidWalls()** to move your robot around for one minute (+/- 5 seconds) without hitting walls. The robot needs to be moving at a minimum of 1/3 of it's maximum speed. (Your robot may drive “backwards” if you want to use the `getIR()` sensors instead of the `getObstacle()` sensors.) The robot should also celebrate after finishing the mission successfully. How it is going to celebrate is up to you, although it must be recognizable. We suggest moving around and beeping at a minimum.

### **--- What's on the robot? ---**

### **Proximity Sensors:**

Proximity sensors are used to detect objects that are close to the robot. The robot has two sets of proximity sensors: one set is on the robot and the other set is on the fluke.

### **The IR sensors on the robot:**

There are two infrared (IR) sensors on the back of the robot (Assuming the fluke is facing forwards). You use the sensors by calling the **getIR(<position>)** function.

Examples:

```
>>> getIR()
[1, 0]
>>> getIR('left')
1
>>> getIR(0)
1
>>> getIR('right')
0
```

```
>>> getIR(1)
0
```

**getIR(<POSITION>)** Returns a integer value in the <POSITION> IR sensor. <POSITION> can either be 'left' or 'right' or one of the numbers 0, 1, which correspond to “left”, and “right”.

IR sensors return either a 1 or a 0. A value of 1 implies that there is nothing in close proximity of the front of the sensor and a 0 implies that there is something right in front of it.

### **The proximity sensors on the Fluke:**

There are three proximity sensors on the fluke: one front sensor and two side sensors. They give different values than the ones on the robot. To use this set of sensors, you need to call **getObstacle(<position>)**.

**getObstacle()** returns a list that contains the values from all three sensors. To get a value from a specific sensor, you can call **getObstacle(<position>)**. <position> can be “left”, “right” or “center”. <position> can also be number 0, 1, or 2, which correspond to “left”, “center”, and “right”.

Examples:

```
>>> getObstacle()
[1703, 1128, 142]
>>> getObstacle('left')
1703
>>> getObstacle(0)
1703
>>> getObstacle('center')
1128
>>> getObstacle(1)
1128
>>> getObstacle('right')
142
>>> getObstacle(2)
142
```

**getObstacle(<position>)** returns a integer value between 0 and 7000. A zero (0) indicates there is nothing in close proximity of the sensor. Higher value implies the presence of objects in front of the sensor(s). Note that the center sensor is the most accurate. The side sensors will detect objects located at about a 45 degree angle, but not those directly to the sides of the robot.

### ***More details about the sensors can be found:***

<[http://wiki.roboteducation.org/Learning\\_Computing\\_With\\_Robots](http://wiki.roboteducation.org/Learning_Computing_With_Robots) – Chapter 5 – Sensing the World> and here:

[http://calicoproject.org/Calico\\_Myro#Reading\\_Sensors](http://calicoproject.org/Calico_Myro#Reading_Sensors)

***Reminder: Your robot needs to be moving at a minimum of 1/3 speed.***

***You may use the timer() function and a for loop to time your robots motion.***

***If you need help with the move functions, go to***

[http://wiki.roboteducation.org/Learning\\_Computing\\_With\\_Robots](http://wiki.roboteducation.org/Learning_Computing_With_Robots) – Chapter 2 – Personal Robots

**Turning it in:**

Be sure to put the lines "from Myro import \*" and " init()" at the beginning of the file (after the required comments). The TA will type avoidWalls() to start your robot moving, so you don't need to include a call to that function in your homework file. Submit your code to the t-square assignment.

**Reminder on collaboration statement:**

This is a pair assignment. Please include your and your partners name at the top, and the names of any other people you collaborated with in your collaboration statement.

**Avoid Walls Grading Criteria:**

<b>Uses iteration correctly</b>	<b>5pts</b>
<b>Code to avoid walls!</b>	<b>5pts</b>
<b>Moves at 1/3 speed or higher for 1 minute</b>	<b>5pts</b>
<b>Celebration in the end</b>	<b>5pts</b>

**You will demo your robot in the next recitation (or at your grading TA's office hours at the helpdesk if you can't make it to recitation) so make sure it's working before your recitation period! Also note that EACH team member must understand how all the code works!**

**Avoid Walls Demo Grading Criteria:**

<b>Moves around for 1 minute, doesn't hit walls:</b>	<b>5 pts</b>
<b>Both team members can explain how the code works:</b>	<b>5 pts</b>