CS 1301 – Fall 2008

Homework 4 – Avoiding Obstacles

Due: Friday, October 10th, at 6 PM EST

(10% late penalty if turned in before Monday, October 13th, at 6PM)

Scored out of 100 points

Files to submit: hw4.py

For Help:

- TA Helpdesk – Schedule posted on class website.
- Email TAs

Notes:

- **Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).**
- **Do not wait until the last minute to do this assignment in case you run into problems.**

- If you find a significant error in the homework assignment, please let a TA know immediately.

Part one --- Robot and the Outside World

This is your first robot assignment. It's time to open the box, take out your robot, connect it to your computer, and play around with it.

There are four 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, and line sensor. We are going to use the robot's proximity sensors for this homework.

Mission:

Your robot will be randomly placed in an arena of size 5 x 3 (Unit: 11 in). You need to write a program to move your robot around for one minute (+/-- 5s) without hitting walls. The robot needs to be moving at a minimum of 1/3 of it's maximum speed. 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.

For more information on the arena, see the posted file.

--- What's in 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 front (or back if you use the end that has the fluke as the front) of the robot. 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 work in a way that's quite different from the ones on the robot. To use this set of sensors, you need to call **getObstacle(<position>)**.

getObstacle() return a list contains all 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. 0 indicates there is nothing in close proximity of the sensor. Higher value implies the presence of objects in front of the sensor(s).

More details about the sensors can be found:
<http://wiki.roboteducation.org/Learning_Computing_With_Robots – Chapter 5 – Sensing the World>

Reminder: Robot needs to be moving at a minimum of 1/3 speed.

Internal Clock function:

To keep track of time, you can use the python internal clock function. To get the current time, call currentTime() function. The function returns the number of seconds past since sometime in the past. [Epoch or Unix time if you're interested.]

Example:

>>> currentTime()
1222374008.360949

To keep track of time, you need to call currentTime() and save the time to a variable (e.g. time). You can get the time that has passed since you last called currentTime() by subtracting time from currentTime().

Example:

time = currentTime()

doing something.....

doing something.....

timePast = currentTime() - time

More detail about the internal clock function:

http://wiki.roboteducation.org/Learning_Computing_With_Robots – Chapter 4 – Sensing From Within

If you need help with the move functions, go to
http://wiki.roboteducation.org/Learning_Computing_With_Robots – Chapter 2 – Personal Robots

Part Two --- Turning it in, and Demo.

Be sure to put the lines from myro import * and initialize() or init() at the beginning of the file (after the required comments). Be sure not to specify the port parameter in your initialize command, such as initialize("com4"). This makes it very time consuming to grade if we have to go into your code and change the com port to the one that works on our specific system.

Reminder on collaboration statement:
This is a group assignment. Each group member need to turn in hw4.py to T-square before the deadline. Please include your name, and all your group members' name in the collaboration statement.

Demo:
Each group (**All members**) need to come to the TA's help desk to demo the program to one of the Tas. ***Ideally you will demo directly to your grading TA***. If that is not possible, you may demo to a different TA, but this may delay your grade slightly as that TA will have to give the grading sheet to your grading TA. All group members will be asked questions about your code at the demo. You may demo to any TA whenever you are finished, as long as your get the demo done before the deadline. We encourage you to demo early! The TA you demo to will fill out the (attached) grading sheet and give it to your grading TA (if they are not the same person).

Be sure your program works as you want before you do the demo, because your grade will be based upon the first Demo! The TA observing your demo *may* give you an extra chance if the robot hits a wall, *at their discretion.*

Grading Criteria:

Demo (See attached grading sheet)	**40pt**
File named correctly	**5 pt**
Uses of internal clock to keep track of time	**15pt**
Uses iteration correctly	**15pt**
Uses IR sensors to detect obstacles (Walls)	**15pt**
Celebration in the end	**10pt**

Written By BoHao Li, Fall 2008

Robot Avoiding Walls Assignment
TA Demonstration Grading Sheet

Group Members: _____

_____

_____

Demo TA: _____

Grading TA (if different): _____

10 pts _____ Robot moved for aprox 60 seconds, *without hitting obstacles*!

30 pts _____ All group members understood and could explain the code.

Total: _____ / 40