Homework 7:

Sonar-Based Obstacle Avoidance using Bug Algorithm(s)

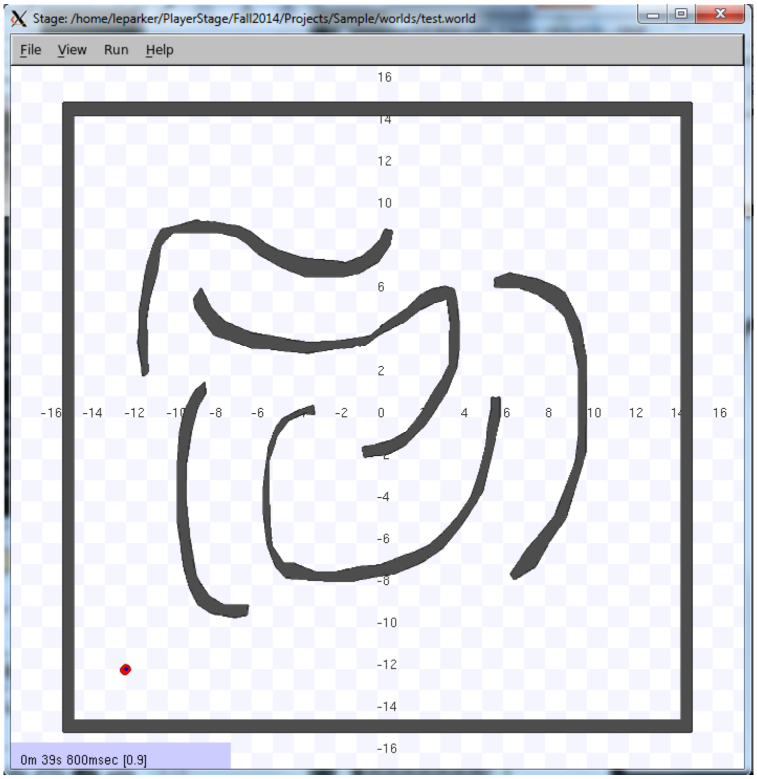
Assigned: Tuesday, October 28
Due: Tuesday, November 11, 2014 at the beginning of class (11:10AM)

In this assignment, you will implement the Bug2 obstacle avoidance algorithm using the sonar sensor. Graduate students will also implement the Bug1 algorithm, also using the sonar sensor.

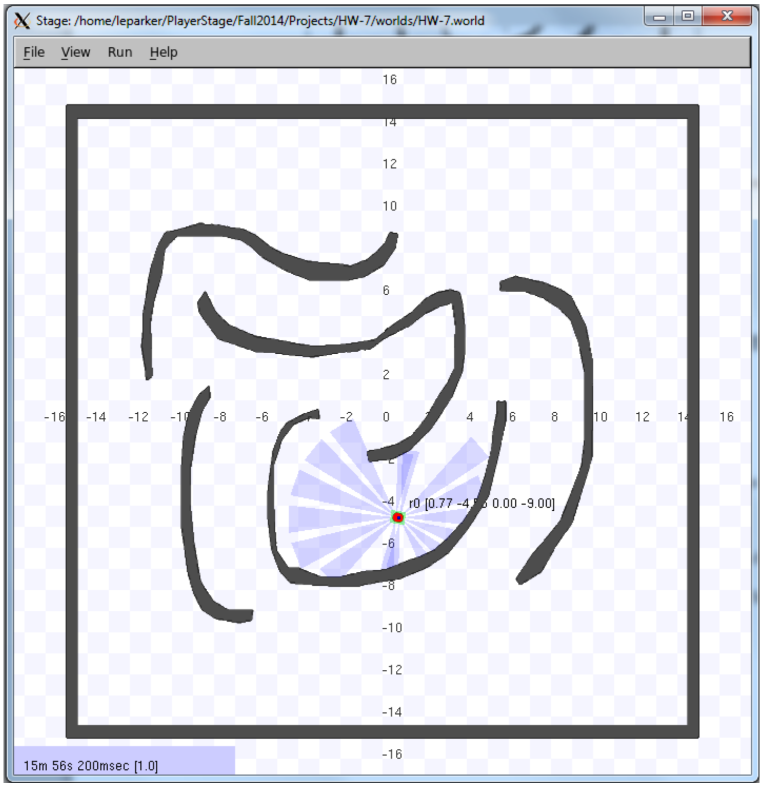
Download the following sample files from the course website:

- HW-7.cfg
- HW-7.world
- randomwalk.cc // Sample code that uses sonar to wander around and not run into things
- wavy-world.png // The environment to use for this assignment

Move the wavy-world.png file to your bitmaps directory. Move the HW-7.cfg and HW-7.world files to your worlds directory. When you start up player with HW-7.cfg, you'll see an environment like this:



The randomwalk.cc sample code shows you how to make use of the sonar. This sample code causes the robot to wander about the world using sonar, avoiding obstacles. Note that when you run this code, you can visualize the sonar sensors, as shown in the image below. This is helpful for debugging purposes. To make it easy on you, this sonar is "perfect", so you don't have to worry about noise in the sonar measurement.



Your software must operate as follows:

- Your code must accept the robot's goal position (in terms of *x*, *y* only) from the user using standard input. Your user interface should make it clear how the user should enter the goal (*x*, *y*) position. The exact format is up to you. You should do some modest error checking to make sure the goal position entered is numeric, and within the bounds of the robot's environment.

- [*Extra credit*] Your code should draw a line (using the technique from HW-6) from the robot's starting position to the entered goal position. This is used to help visualize the algorithm's behavior.

- [*Everyone*] Your code must implement the Bug2 obstacle avoidance algorithm using the algorithm in the text (section 6.4). You don't have to use the exact code provided in the text, however. Just be sure the algorithm is implemented as illustrated in Figure 6.15.

- [*Graduate Students Only*]: Your code must also implement the Bug1 obstacle avoidance algorithm, using sonar, as described in the text. Your two algorithms should be in the same program file; just prompt the user for which algorithm should be executed.

Testing your code: Your Bug2 (and, for grad students, Bug1) obstacle avoider should work for the robot starting at the position (-12, -12, 45) (i.e., the position specified in the HW-7.world file), and going to any goal position given, as long as the goal is reachable. (It is OK if your robot/algorithm runs forever if it is given a goal position that is unreachable.) Test your code for lots of different goal positions. Then, create screendumps of your robot's path to the following specific goal position:

(11, 11) // which is in the upper right corner of the environment

WRITE UP THE FOLLOWING (written up in a single pdf file called *yourlastname*-HW-7.pdf):

a) A discussion of issues you ran into in getting your algorithm(s) to work properly (about ½ page). I only want algorithm-specific discussions about the robot behavior.

b) [*Everyone*] Screenshots of your robot executing the Bug2 algorithm from the provided starting point (-12, -12, 45) to the goal position (11, 11).

c) [*For graduate students*] Screenshots of your robot executing the Bug1 algorithm from the provided starting point (-12, -12, 45) to the goal position (11, 11).

SUBMITTING YOUR HOMEWORK:

Place all your files in a single directory. These files should include:

- Your written answers to the questions above, plus screen dumps (images) as requested (i.e., the file "yourlastname-HW-7.pdf")
- Your makefile, called "makefile" or "Makefile"
- Your robot control code, called "yourlastname-HW-7.cc".
- Any additional include files you created (called whatever you want them to be called).

Remove all other unnecessary files, and tar or zip them up. Compress if needed. Submit to BlackBoard.