Mini-Assignment 1:
Getting Started with Player/Stage

Assigned: Tuesday, October 23
Due: Friday, Nov. 2 at 23:59:59

[This assignment is easy!! This assignment counts the same as 1/4 of a typical project assignment].

The purpose of this mini-assignment is to help you become familiar with the Player/Stage robotic simulation environment we will be using for some of the assignments. This exercise involves setting up your simulation environment the simulation software and experimenting with various aspects of the system. You should learn how to run the simulator, how to run robot code, and the functionality of various Player/Stage files. You should approach this homework as an opportunity to explore and learn about Player/Stage.

> **A note on collaboration:** For this assignment, I highly encourage discussions amongst yourselves for the purposes of helping each other learn about Player/Stage, and/or to help each other get everything set up and running. Talk freely, and help each other as needed. However, each person must individually prepare all the material to be turned in. No sharing of output or textual descriptions (as outlined below) is allowed.

This simulation system operates under Linux (any of the Hydra machines in the CS lab should work) using C, C++, and Python. (It also works with other languages, but not all the languages have the same supporting examples or are as well supported. The examples we provide are in C++.) Turn in the information noted below, clearly numbering the material according to the outline below.

1. **Background Reading:** "Most Valuable Player: A Robot Device Server for Distributed Control", by Gerkey et al., is located on the course web site here:

   http://web.eecs.utk.edu/~parker/Courses/CS494-529-fall11/Handouts/Player-Stage-paper.pdf.

   This paper provides some background on how the Player/Stage driver and simulator is set up. This is just for your information, in case you are curious. Note that documentation on Player/Stage is available in the public domain, here:
   http://playerstage.sourceforge.net/index.php?src=doc

   You may also find the following manual to be helpful:
   http://www-users.cs.york.ac.uk/~jowen/player/playerstage-manual.html

   Make use of these resources!

   (Nothing to turn in for this step.)

2. Set up your simulation environment

- Follow the instructions in the Player/Stage Getting Started Guide, located at:

  http://web.eecs.utk.edu/~parker/Courses/CS425-528-fall12/Handouts/PlayerStageGettingStarted.html.

- At this point, you should be running the robot using simple.cfg and the laserobstacleavoid program (as outlined in the Getting Started Guide).

- Explore various menu options on the pull-down menus of the Stage simulation window. For example, note that you can turn on a trace of the robot's motion using a pull-down menu in the stage simulator (called footprints), and that you can create a screen dump of the simulation window using another menu option. You can also turn on and off various sensor displays. Also note that you can drag the robot around using your cursor.

- ***#1 TO TURN IN: A screen dump of your robot's motion (including the trace of its motion), after running for a few minutes using simple.cfg and laserobstacleavoid.***

3. Experiment with other sample programs using everything.cfg configuration file

- Re-start Player/Stage using the everything.cfg configuration file.

- Now experiment with different programs that are in the "examples" directory (such as sonarobstacleavoid, randomwalk, etc.). Not all the programs will work with any arbitrary configuration file, since the configuration of sensors has to match the sensors expected in the robot program.

- ***#2 TO TURN IN: Pick any two other working programs (besides laserobstacleavoid), and for each, turn in a screen dump of your robot's motion (including the trace of its motion), after running for a few minutes. Be sure to note which program generated your screen dumps.***

- Now, run Player/Stage with everything.cfg and the randomwalk program. Study the robot's behavior, and examine the code for randomwalk.cc. Make use of the Player/Stage documentation as needed to understand (at the high level) what the program is doing.

- ***#3 TO TURN IN: In descriptive text, based upon your examination of the code and the behavior of the robot in the simulator, explain how the randomwalk program works.***

- Now, study the everything.cfg configuration file. Make use of the Player/Stage documentation as needed to understand (at the high level) what the configuration file is doing.

- ***#4 TO TURN IN: In descriptive text, based on your examination of the everything.cfg file, explain what the purpose is of the .cfg file, and (generally) what it contains.***

4. **Experiment with other bitmaps**

- Note that everything.cfg specifies the use of the worldfile "everything.world". Examine the everything.world file. Make use of the Player/Stage documentation as needed to understand (at the high level) what this file is doing.

- ***#5 TO TURN IN: In descriptive text, based on your examination of the everything.world file, explain what the purpose is of the .world file, and (generally) what it contains.***

- Note that the world file specifies the robot's environment using a bitmap from the worlds/bitmaps directory. Change the everything.world file to specify a different robot environment (pick one from the bitmaps directory), and re-run player with everything.cfg. This time, the different robot environment should appear.

- ***#6 TO TURN IN: A screen dump of the new environment using everything.cfg.***

Undergraduates and Graduates: Turning in your project

Put all of the required material (#1-#6 above) into a single pdf file. Then, submit the pdf file using the UTK course blackboard.