<div align="center">

CMSC 143: Object-Oriented Programming with Robots

Lab 9: Online Robots

Due November 6, 2014

</div>

Online robots like the telegarden[1] combine the Internet with robotics to provide telepresence. In this lab, you will create a Python program that when run, builds a webpage describing the state of your robot.

You will write a Python program that generates HyperText Markup Language (HTML) code which you can then upload to a webserver or look at locally in a web browser. You are encouraged to post the resulting HTML file to your student web space on Bard's student webserver[2].

First, you should read the first few HTML tutorials provided by the w3schools.com[3]. HTML is a markup language, meaning that along with the actual content (the data) the text file also also contains annotations of the content (meta-data). In HTML, the annotations, or tags, are predetermined commands that describe the structure of the document and also how it should be displayed (e.g. `body`, `a`, `img`). The HTML tags are enclosed in angle brackets and the tags are often found in open/close pairs.

Guidelines for the webpage:

1. Robot: You must show the robot's name and some of the robot's sensor values (light, infrared, battery).

2. HTML: You must use a heading, a link, an image, a list, a table.

Beyond those specific guidelines, you should be as creative as possible. You might include or provide links to some of the programs you have created.

Learning Objectives

  ○ Work with File I/O   ○ Use HTML   ○ Create a Meta-Program

Deliverables

Submit an zip archive of a directory including your program (.py), your webpage (.html), and other supporting files via moodle.

```
cmsc143_lab9_LASTNAME/
      lab9.py
      lab9.html
```

Notes

You should create helper functions that generate HTML as strings for certain elements. For example, you might create a function `createUList(l)` that generates the HTML necessary for displaying each element of the list `l`. Also, if you are interested in having the web page automatically refresh (to continually show the status of the robot, for example), you can use the following tag inside the `HEAD` section of the HTML page: `<meta http-equiv="refresh" content="1" />`. And to prevent the browser from caching images use: `<img src = "FILENAME.JPG?v=CURRENTTIME">` where `CURRENTTIME` is changing based on the `currentTime()` function.

---

[1]http://www.usc.edu/dept/garden/
[2]http://inside.bard.edu/bits/software/ftp/
[3]http://www.w3schools.com/html/html_intro.asp