CMSC 143: Object-Oriented Programming with Robots

Lab 6: Online Robots

Due October 13, 2016

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. The Python program will generate HyperText Markup Language (HTML) code which you can then upload to a webserver or look at locally in a web browser.

First, you should read the first few HTML tutorials provided by the w3schools.com[2]. 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. The generated web page should be validated HTML[3].

Learning Objectives

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

Deliverables

Submit an electronic copy using moodle and a hardcopy of your python program. Your program should have your name, email, assignment description, the date, and collaboration statement at the top of the file as a comment. Your submission should be a zip file that expands to a folder with at least these two files:

```
cmsc143-lab6-LASTNAME-FIRSTNAME
      lab6.py
      lab6.html
```

Helper Functions

You should create helper functions that generate HTML as strings for certain elements. For example, you should create a function `createH1(text)` that generates the proper HTML, for example, `createH1('Title')` would return `'<h1>Title</h1>'`. Try and also create functions for lists or tables, e.g., `createUList(l)` generates the HTML necessary for displaying each element of the list `l`, but these are a bit trickier.

```
assert createH1('My Robot HomePage') == '<h1>My Robot HomePage</h1>'
assert createLink('http://bard.edu', 'School') == '<a href = "http://bard.edu">School</a>'
assert createHTML(createBody('My Page')) == '<html><body>My Page</body></html>'
```

Finally, once you have the entire HTML page as a string, use the `write` method to dump the string into the file (`lab6.html`). Use assertions to sanity check your code.

[1] http://www.usc.edu/dept/garden/
[2] http://www.w3schools.com/html/html_intro.asp
[3] https://validator.w3.org/#validate_by_upload