Advanced Programming (CSE 399)
Homework Assignment 7

Due Friday, March 4, at noon

Background: If you haven't done much HTML programming, you may need to do a little background reading on how things like input forms work. There are numerous good sources for this on the web. Also, if you've never played the original `adventure` game, you owe it to yourself to check out

`http://en.wikipedia.org/wiki/Colossal_Cave_Adventure`.

Preparation:

1. On one of the linux lab machines, grab the tarball for the sample WASH program from the Schedule page, unpack it somewhere, `cd` into it, and do `make`. The result should be three executable files, called `ex1`, `ex2`, and `ex3`. Point your browser at

   `http://fling-l.seas.upenn.edu/~YOUR-USER-NAME-HERE/cgi-bin/ex1`

   to check they are working.

   The provided `Makefile` will only work on linux-based `eniac` machines. If you want to do the assignment on another machine, you'll need to do your own installation of Wash (probably pretty easy, if you already installed GHC) and run your own web server (trivial on OSX, easy on other Unixes, no idea about Windows).

2. The provided tarball includes a subdirectory `Wash` containing the entire Wash/CGI distribution. Spend a few minutes exploring, to get a sense of what is there.

3. Copy two or three of the files from `Wash/Examples` into the top-level directory (to get the right Makefile) and try compiling and running them.

4. Spend a few minutes browsing the Wash/CGI on-line documentation (in the `doc` subdirectory under `Wash`). Concentrate on the top-level module `GuaranteedCGI`. (Note that Wash provides three top-level interfaces: `CGI` is the simplest, `GuaranteedCGI` uses the Haskell type system to guarantee quasi-validity of generated HTML, and `CGIXX` provides extra functionality but depends on non-Haskell-98 language features. I recommend using `GuaranteedCGI`.)

Assignment:

1. Point your web browser at

   `http://fling-l.seas.upenn.edu/~bcpierce/cgi-bin/Adventure.cgi`

   and experiment with what you find. (Can you figure out how to get to the node that says "Hooray, you win"?) Note, in particular, what happens if you enter a command that is not available, or if you press the "Go" button without entering a string in the text field. (Try moving the mouse over the red question mark.)

   Use WASH to write a CGI script that duplicates this functionality as closely as possible.

   Although the main purpose of this assignment is getting experience with the features of WASH, it doesn't hurt to pay attention to modularity at the same time. Break your solution up into at least two modules, with clean, well-documented interfaces.

2. **Extra credit:** Enrich your program with additional features of your choice. Examples:

   - Add some more interesting content.
   - Make the generated web pages look more snazzy.
   - Display an alternate (shorter) description when visiting a node that has been visited before.
   - Add the ability to place objects at nodes, some way of picking them up, and some way of using them (for example, certain directions from a given node might only be traversible if the play is carrying some object.)
   - Make the interaction graphical (see `http://www.kingdomofloathing.com` for some possible ideas).

Submission:

- Put an executable for your final program in your `cgi-bin` directory on `fling-l` (or any other publically accessible web server) and email its URL to `bcpierce@cis.upenn.edu`. Attach your source code to this message too.
- If you choose to do the extra credit part, write me a *separate email* listing the extensions you've implemented and explaining how to exercise them.