CMSI 370

INTERACTION DESIGN

Fall 2010

Assignment 1209

The readings aren't extra credit, but the programming work is. If you choose to do the extra credit assignment, you'll get a little taste of command/natural language/speech programming.

Not for Submission

1. Read Chapter 7, and skim through Chapters 8 and 9, in Shneiderman/Plaisant.
2. Read the two "The Next UI Breakthrough" columns by Don Norman (the "Physicality" one was distributed a few weeks ago).

For Submission (Extra Credit)

Download the source code to CMU's Sphinx 4 (*http://cmusphinx.sourceforge.net/sphinx4*), then follow the instructions for setting it up and building it.

Using Sphinx 4's *HelloWorld.java* demo program (and its associated files) as a basis, implement a simple "shell" for a speech-driven, command-based text adventure game. Call the program *Dungeon.java*, and have it do the following:

- Perform any necessary initialization.
- Wait for the user to speak the following commands, displaying the converted text each time:
  - *walk north*, *south*, *east*, or *west*
  - *inventory* or *show me my stuff*
  - *take* one of some collection of objects
  - *drop* one of the same collection of objects as above (i.e., if an object can be taken, then it should be possible to drop it)
  - *look at* one of some collection of objects, or just *look around* by itself (this must be a superset of the objects than can be taken)
  - *quit* (this ends the program)
- You don't have to implement an *actual* game; just set the program up to recognize the above spoken commands. You'll need to know how to define a BNF-style grammar (ooooh, theory tie-in!) using the Java Speech API Grammar (JSGF) format. All necessary information can be found in the above Sphinx 4 web site. The December 2 class will have also demonstrated some version of the above tasks.

Extra Extra Credit

If you are so moved, go ahead and implement a baby text adventure game **:)** You'll need to program a set of locations or places through which the user can travel, as well as a set of objects which the user can look at, and possibly take. The program should enforce the world that you define: that is, the user should not be able to walk to a non-existent place, nor walk in a direction where there is no predefined place. The user should only be able to look at or take objects that are actually in the current location; taken objects should show up in the user's inventory; and only objects in the inventory can be dropped (with dropped objects then showing up at the current location).

How to Turn It In

For full extra credit, this work needs to be submitted before class on December 9. Late submissions will get half extra credit.

You may assume that I already have the base Sphinx 4 software; just send me your *Dungeon* source files.