COGS 211: New Behavior Lab #2

Guidelines

1. At the conclusion of *last* lab you should have accomplished the following:
    a. Made a decision about the behavior you want to create
    b. Identified the key components of that behavior
    c. Worked out a rough flow chart describing how information will be gathered, processed, and used to choose the actions that will demonstrate your behavior.

2. *Today* you should attempt to accomplish at least the following
    a. Complete an in depth analysis of exactly what problems must be solved in code in order to implement all of the components of your behavior.
    b. Render that analysis as pseudo code (see Jones for examples of what that looks like)
    c. Begin the process of turning your pseudo code into working code

Tips for managing workflow[1]

1. Do NOT attempt to build your new behavior by tweaking or modifying a copy of the existing code. Instead, begin a brand new project and copy material from the sample code as you need it, and then modify it in the new project. Why do this?
    a. Because it will force you to understand all of the elements that are necessary for YOUR code to work. Otherwise you will not properly appreciate what parts of the sample code are and what parts are not relevant to your own behaviors. You have to explain this in your lab report, so make it easy on yourself from the beginning.
        i. Note that this will also be a very effective way of learning how to code. Simply tweaking existing code is less informative.
        ii. It will also force you as a group to talk to each other about what components need to be added and why, another effective learning device.
    b. This also sets you up to follow the next piece of advice more effectively.
2. Decide on a naming convention for your files that guarantees that you know what the most recent functioning version of your code is. Here is one example of how to do that:
    a. 171106_1333_TeamName_W.c

[1] This is especially useful for those of you who have little programming background but is also useful for those of you not used to working in groups on a coding project.

b. This says that the file was last updated on the 6th of November, 2017, at 13:33 (1:33 p.m.) by your team and that it was working (W).

c. You may use a different naming convention if you devise one that works better for you. The key is to make sure that *everyone* on your team understands the code in exactly the same way. You might consider practicing on a trial run to make sure you all give the same name to an imaginary file.

d. Some of you with more experience may be tempted to use GitHub for file management. Do this ONLY if every member of your team is familiar with how it works, and ONLY if you have all agreed on your file management rules. I have seen very experienced programmers go badly off the rails by misusing GitHub. It's a great system, but use it only under the conditions described above.

3. Once you have ANY code that compiles and runs, make sure that you save it as a working file BEFORE you make any additional changes. I have stopped counting the number of times I have seen people get something working and then, in a moment of unthinking enthusiasm, immediately begin tweaking the code to take it to the next level, only to discover that in so doing they have broken the working code and can't remember what was changed. Indeed, I've made that mistake myself, and I know the pain of loss it causes. ***Don't be that team.***

4. Note that you should keep your own notes individually in a digital or paper file on exactly what every line of the code does. Do not clutter your file with group code. Normally you would comment as a team as you go, and you can do that in a very minimal way so that you don't lose track of things, but remember that you will each be responsible for providing a commented version of the code as an appendix to your final lab report. The coding can be done as a group, but the commenting must not be. This is how you will show that you understand what the code does. Be careful, therefore, to make sure that you final commenting is in your own words and not shared words of the group.