Final Project Information

There are several due dates for this assignment. The project proposal is due by 10pm on Monday February 26. Project demos will occur during our finals period from 8:30-11:00am on Monday March 12. The final project is due by 5pm on Wednesday March 14.

1 Goals

The goal of this assignment is to get experience with all the concepts discussed in class while implementing a more complex project.

2 Project Details

You have two options for a final project: create a computer game that incorporates some level of computer intelligence (I'll tell you more about what that means), or create a data analysis and corresponding visualization.

1. **Choice 1:** A computer game that incorporates some level of computer intelligence.
   - Your game can be entirely text based, or you can use Turtle graphics, graphics.py (a simple object oriented graphics library which you'll use for the next homework), or cImage.py if you'd like to make it more visually appealing. **No graphics are required**, though; I'm much more interested in seeing how much you've learned about how to implement the algorithms that you want to use.
   - Some possible games to implement: Scrabble, Bagels, Dots and Boxes, Battleship, Go Fish, Bejeweled, Connect Four, Mastermind, etc. There are many other possibilities that are not on this list.
   - Some games to be wary of: a text-adventure game (these can be done well, but they usually devolve into a morass of nested if statements that aren't interesting to write or read); Hangman, Tic-Tac-Toe, Pig/Hog, etc. (too simple); Go, your favorite German board game (too complex).
   - The requirement of "some level of computer intelligence" means this: the computer should play against you or manage your game in some kind of intelligent way. You don't need the computer to be a world-class player at your game (or even close!) - that's what CS 321 (Artificial Intelligence) is for - but you should make an effort for it to do something reasonable.
2. **Choice 2:** Data investigation/visualization.
   - If you choose this option, you should select the dataset of your choice and come up with a well-formed, testable hypothesis. For example, maybe you want to collect some census data and public-health statistics, then compare all that to some theoretical epidemiological model. Or maybe you'd like to process a bunch of ancient texts written in Latin, looking for allusions among them. Or perhaps you've got a giant database of protein interactions in human cells, and you want to try to infer something about the network of interactions as a whole.
   - The software you write should analyze the data, collect it autonomously (if that makes sense for your data source), and present the results in a useful and non-trivial way (e.g., a visualization). Matplotlib (which you saw on hw07) is a really useful module for visualizing data. There is a lot more you can do with it than what we saw in the assignment.

- Your analysis should incorporate a fairly large dataset (i.e. something that you couldn't reasonably analyze by hand), and your final project should also include a short writeup (2-3 pages, single-spaced) that discusses the background of your problem, your hypothesis, your data sources, how your program addresses your hypothesis, the results, and the conclusions you draw from these results.
- Your visualization should also be influenced by user input in some way: for instance, the user could indicate how many years of climate data to include, or determine how many graphs to display, etc.

2.1 General Requirements

Regardless which of the two options above you choose, the code that you submit should demonstrate the following:

- Good Object Oriented Programming (OOP) practices. In short, in your project you will need to build and utilize multiple classes of objects.
- Appropriate use of functions/methods rather than similar (or identical) repeated code.
- Documentation standards as outlined in the Advanced Style Guide.
- Complete documentation of any outside resources that you use. Specifically, read the section on academic honesty below and review the course honesty policy in the syllabus for full details of my expectations.

Additionally, I require that you submit a text file called README.txt. Specifically this file should include the following information:

- A description of your program and its features.
- A brief description/justification of how it is constructed (class organization, how data are stored, etc.)
- A discussion of the current status of your program - what works and what doesn't, etc.
- Instructions for running your program. I will use these instructions when running your program, so make sure they are clear. If I don't know how to run your program, it's very hard to evaluate it!

2.2 Partners

For this project, I will allow you to select your own partner. Once you have found a partner, make sure that you both agree to work with each other (and not with anyone else). I will allow individual projects if you have a good reason that you wish to work alone. I very strongly encourage you to work in a partnership; it will let you accomplish more and, I hope, have more fun. If you want a partner but haven't been able to find one, please try posting to the class questions forum or come talk to me. We'll make sure that everyone who wants a partner has one.

Unlike with pair-programming assignments from the course so far, you are not both required to work on all aspects of the project together, but you are all expected to do your fair share. This means you can split up parts of the work, but will probably want to come together when integration your different pieces of work.

3 Deliverables

Project Proposal

Due date: Submitted on Moodle by Monday, February 26, 10PM. Only one of you and your partner needs to submit a proposal.

You must submit a proposal for your project. This proposal should include

1. Your partnership (or an explanation of your plan to work alone)
2. Which of the two categories your proposed project falls into.
3. An overview of what you're planning to do. This should include a list of pieces (e.g. classes) that you plan to implement, in order to create your project. If you are doing the game option, you should include an overview of how you think the computer intelligence piece will function. You could think of this proposal as a plan of attack for your project, so that you can think through what steps you will need to take to make your project work. Writing a good proposal will make it easier to actually carry out your project.

Let me know if you have questions about the rules or suitability of a particular idea. I (as opposed to the course grader) will provide feedback for all submitted proposals. You may certainly come talk to me about your thoughts about the project early to make sure that you haven't picked something too easy or too hard. The biggest trap on this assignment is picking something too complex and not having the time to finish it.

Final Project

Due date: Submitted on Moodle by 5pm on the last day of finals, Wednesday, March 14. No late projects may be accepted without an extension from the dean's office.

Create a directory that has all of the files for your final project - make sure **all** of the files are in that directory. Additionally, in this directory, include a file called README.txt (described in more detail above). Zip up this directory, and upload it to Moodle. Additionally, fill out the final project partner survey on Moodle if you worked in a partnership; this allows me to know if you both contributed or if someone did all of the work, and to set grades accordingly.

Final Project Demos

During our final exam time slot (**8:30-11:00am, Monday March 12**), we will have a demo session. **Your projects don't have to be complete at this point, but you should make sure you have something to share with me and with your classmates.** This is a great opportunity to show off your work, see what others have done, and get formative feedback to help you make your project as awesome as possible by the due date. We'll likely spend about the first half of the exam period on demos, and in the second half, I'll stick around for informal office hours in case you want help debugging/figuring out how to do something/etc.

4 Grading

Your project will be graded based on the following criteria, in decreasing order of importance:

1. **Correctness:** How cleanly does the program run? Is it bug free? Does it crash?

2. **Complexity:** How complicated is your game? How complex is the computer intelligence?
3. **Style:** Is the code well organized, readable, and well documented? Are you making classes in cases where it might be helpful? Are you avoiding duplicated code? (Keep following the style guidelines on Moodle!)

Correctness is the most important factor. Let's say you choose the game option. A simple text-based program that runs correctly, is error-free, and achieves the above specifications will get a better grade than an incredibly smart chess program that crashes and produces unreadable output. (Of course, if I'm comparing side by side two programs that run perfectly, the one that accomplishes more will receive the better grade. But that's a secondary concern.)

To reiterate: It is most important to submit something that works. Set modest goals and achieve them first, then enhance your program later. If you choose a game, only work on getting computer intelligence working after you have the basic game working with human player(s).

On academic honesty: You must submit a program that you write yourself/yourselves. You are more than welcome to obtain help from other students in getting ideas and in debugging your code, but you must write your own program. No one else should type in any code that you submit. You may not use Python code for your game from the Internet or from other sources. (Ask me first if you'd like to use a module that you find online.) You must acknowledge any sources of assistance (for example, if you founds some useful methods for the list class (which we didn't discuss in class) using the online Python documentation - cite the documentation website in your comments! Remember, I must report suspected violations of academic integrity to the dean, and if you are found responsible, the penalty is potentially failing this class. It's not worth it.

5 Last Piece of Advice

Start early, ask lots of questions, and **have fun**! This is a chance to really apply everything you have learned this term to a cool project of your choice. You've come a long way since week 1 - you should be proud of all that you have accomplished. Layla, the lab assistants, and the prefect are all here to help you succeed - don't hesitate to ask for help if you're struggling!