Advanced Topics in Comp Sci — Independent Project

ASSIGNMENT OVERVIEW

In this final, independent assignment you'll be designing, creating, making, and/or writing a project of your own choice.

This assignment is worth 100 points and is due at your GitHub account at 23:59:59 on the date given in class.

BACKGROUND

This class has been focused on algorithms and data structures, and the study and/or implementation of a project that involves using an algorithm based on one or more of the data structures that we've covered—or even some that we haven't—would be a perfectly appropriate choice for a final project in here.

It may be that there are other technological or computer-based challenges that you might wish to investigate, however.

PROJECT SPECIFICATION

Develop a project centered around one of the following challenges, or design your own project and propose it to the instructor for approval.

a. Implement the RSA Algorithm
b. Demonstrate the Collatz Conjecture / Syracuse Sequence
c. Build a computer-based hardware project: "arduino servo pumpkin eyes," for example.
d. Demonstrate Dijkstra's algorithm
e. Study and demonstrate buffer overflows
f. Study and build a machine learning (ML) facial recognition system
g. Participate in an open source project
h. Implement a *minimax* algorithmic solution to a game

The development of your project depends in part upon your choice of subject: a hardware-based project will require the use of hardware, a software-based project will require developing software, etc. Your project must be a "new" project for you, or a previous project that you want to extend in specific ways. Previously completed projects can't be submitted for this assignment.

Your project must also consist entirely of work that *you* yourself have done. Code-based projects will consist of code that you have written yourself, with possible use of a code library, or by consulting with an algorithm reference or demonstration code snippet where needed. References to this external code must be cited in both your internal comments/documentation and in the `README.md` file accompanying your project. *Code that you submit as yours for this project must not be code that others have written.*

All projects submitted for this assignment will require extensive documentation as described below in the *Deliverables* section.

It is expected that your project will require 8-15 hours of work in and outside of class.

DELIVERABLES

`ATCSIndProj.git` - a git repository hosted in your GitHub account

The materials for your project should be collected and well-organized in a directory called **ATCSIndProj**, along with any other supplemental materials as dictated by your source of project.

Contents of this directory:

- There must be a `README.txt` file containing appropriate information regarding your project. See the course website for examples of such files.
- There will probably be a `src` directory with files containing the classes and/or programs that you wrote for your project. If this directory also contains third party libraries/dependencies for your project, make sure they are clearly identified as such, and documented in your source code and `README.md` files.
- There may be other directories as needed to document the work that you've done on your project

To submit your assignment for evaluation, make a final commit to your project in your local project directory, and push that commit to your GitHub account before the deadline.

ASSIGNMENT NOTES

- Perhaps the single most challenging aspect of this assignment for some students is identifying a project to work on. You may be (justifiably) concerned about picking something that's too easy, or selecting a project that turns out to be unmanageable in the time we have remaining. Knowing in advance the *scope* of your problem is often difficult, and sometimes impossible. Don't waste time fretting about whether your problem will be too easy or too hard. The most important thing is to identify something of interest that you can work on, and get started.

- Propose your topic of study to the instructor, perhaps in conversation, and certainly by the formal mechanism provided (a shared Google sheet).

- If your proposed project is too easy, you'll need to enhance it in some way to make it more challenging. Writing a program that allows two people to play Tic-Tac-Toe on the computer is easy; writing a program that has the computer play Tic-Tac-Toe against itself, and "learning" to optimize its playing as it wins and loses is more appropriate.

- If your proposed project is too difficult, you'll need to find a way to wrap things up as gracefully as possible. A working version of a simple proof-of-concept of your program, for example, is better than one that doesn't work at all. Extensive documentation of the real challenges you encountered is especially important if you are turning in a not-yet-working project. Even better, of course, is finding a way to get it working.

- Managing your work on the project will require some degree of planning, and you almost certainly already know that projects—both software and hardware—*never* take less time than you'd planned, and rarely take as much time as you'd planned. They take *more* time than you'd planned. With that in mind, why would you procrastinate? *Get started now!* (See the *Getting Started* section below for ideas on what to do first.)

- **Daily commits.** The expectation is that you will have at least one commit each day of class. You may have more as a result of working on the project outside of class. Evaluation of your project will in part be based on whether or not you demonstrate progress on your project, as revealed by your daily commits.

GETTING STARTED

1. Identify for yourself what you think you'd like to focus on, and get approval for your project (via the shared Google sheet) as quickly as possible.
2. Identify what your plan is. You can use paper and pen for this (take a picture for documentation later), or a text file with a preliminary schedule, or a calendar. Build in time in your schedule for unanticipated setbacks, because those will certainly happen.
3. If you are having a hard time getting started—the magnitude of your project is overwhelming, perhaps—begin by accomplishing a few small tasks. Start typing your `README.md` file. Do some research on your topic and copy down some URLs for later reference.
4. Hardware projects begin with assembling materials. Find out what you need and get things ordered for delivery.
5. Software projects begin with a source code file and a `git init`.