Final Specification

Due Friday, April 17 at 5PM (Friday, April 17 at midnight for Extension)

Overview

Last week we had you submit a rough outline of the implementation details of your project. This week we want you to fully flesh this out and begin writing some code.

Before submitting this final specification, you should meet with your final project TF. If they haven't contacted you by Wednesday, April 15, please email Ben and Allison.

You have already thought about how to organize your project and created a rough outline of the technical specification. You will now be formally specifying your modules by creating an interface for each. You should then begin writing the actual implementations based on these interfaces.

Be sure to consider any advice you recieved from your TF after the last checkpoint concerning the draft specification.

Requirements

- Code: write complete interfaces and begin implementation of key functionality
- PDF: write and submit final specification
- Other: set up version control system, meet with your TF

Detailed Description

Signatures/Interfaces

For each of the major components, you should write down a full interface or contract for that component. Things to think about:

- What types or abstractions will you provide? How will these abstractions reduce the conceptual complexity of your project?
- Which values and functions will you expose? Which will you hide?
- What properties should clients of the component respect (e.g., all input values

must be of a certain format)?
- What properties will the component ensure?

If you are coding in a language like OCaml, then you should specify the signatures for each component in the language itself. Take a look at the online documentation of some signatures or interfaces for your implementation language if you're unsure what this section should look like.

Advised but optional: write out pseudo-code for the most important parts of your specification (your algorithm or data structure). This will help you fully understand how things should work without having to worry about the quirks of whichever language or tools you are using. This will likely make implementation easier.

Modules/Actual Code

Begin writing code for the most important parts of your project. At this point you have the skeleton of your project, and you should begin writing the code most fundamental to the success of your project. All of your projects are very different, so we cannot tell you specifically what you need to do. Use the writeup as a place to show us the progress you have made on your project.

Timeline

As you begin writing code, it is useful to set up concrete goals and checkpoints for yourselves. Please include a roadmap to your completing the project, with 4-6 bullet points per week.

Progress Report

Please include any additional information regarding your project's design. Also, use this space to convince us that you are actually making progress on writing code. The staff will not blindly search through your source files. Either include your progress in the writeup or point us to one or two files that demonstrate your progress.

Version Control

We would like you to set up a version control code repository for your group. The best ways to do so include setting up a new git repository on code.seas or using a commercially available service such as Github or Bitbucket. Dropbox, email, and Google

Docs are not acceptable version control systems.

Please let us know what version control repository you set up in your progress report. Also, feel to give your TF read access as a way to keep them posted about your progress.

Submission

Email your writeup (as a PDF) to your final project TF. Only one person should submit for each group. The subject of this email should be "CS51 Project: Technical Specification". Please include a zip file containing the code you've written thus far.