Final Project Specifications and Timeline

Now that we're covered recursion (an algorithmic abstraction) and started in on pointers, dynamic memory, and linked structures (low-level concepts necessary to support custom abstractions), you're in a position to design, implement, and showcase an application of your very own. The final project is intentionally open ended, as it's your opportunity to demonstrate you've mastered the material and synthesize a nontrivial codebase to do something you couldn't have done prior to taking this class.

The amount of code you need to write will depend on the problem you choose to work on, though it needs to be at least as large as something like Assignment 3's Boggle, Assignment 5's Priority Queue, or Assignment 6's Huffman. At a minimum, your final product needs to rely on recursion, custom data structures, or both. You're otherwise permitted to build anything you want, and you're even welcome to program in a language other than C++, provided you know the language well enough to be productive and work independently of the CS106X infrastructure you otherwise get by using C++.

Project Ideas

You're welcome to work on anything you want to, but here's a short list of ideas, just in case you're having a tough time brainstorming.

- Implement some human-versus-computer game (Othello, Chess, Checkers, WordSearch, Scrabble, Go, etc.) that requires some form of minimax implementation for the computer player. This would likely require a small amount of graphics programming, but the heart of the project would be coming up with heuristics to ensure the computer can more or less annihilate the human every time.
- Implement a well-defined interface in many different ways—much as Assignment 5 will ask you to—and present timing and memory resources to illustrate the tradeoffs that come with different internal implementations.
- Build a miniature version of one of your favorite consumer products—Facebook, Twitter, Pinterest, Instagram, WhatsApp, etc. This would require you learning a bit about networking and how to write a server, but it's well within the scope of what you're capable of doing right now.
- Explore http://openframeworks.cc/ to synthesize something beyond the reach of what can be easily built using the CS106X graphics libraries. (One idea: build a platform that allows you to control stage or show lighting, e.g. Tree of Ténéré at Burning Man 2017.)
- Replicate the results of any of the research talks you've seen while in CS106X or elsewhere. In particular, if you're a graduate student in another department, then you might verify some result of something important to your research.

- Build a simulation that taps your expertise in physics, chemistry, biology, automata theory, logic, or mathematics. (Two ideas: build a theorem prover using first order logic, or build an animated simulation illustrating how new bitcoins are mined.)
- Build a simple database that supports an SQL-like query language, or some other query language optimized for the types of data being stored.
- Dabble in natural language processing by downloading and/or scraping large data sets and building a classifier of some sort. The data might be restaurant reviews, real estate listings, election results, newspaper articles, audio files, music scores, Wikipedia pages, 18th century novels, or whatever you can think of.
- Do whatever else you can think of instead: data analysis using neural nets, SVM, etc., something to do with encryption, security, blockchain, or perhaps build a recommendation engine in the style of Amazon or Netflix.

Timeline

Here's a list of deadlines you need to meet between now and the end of the quarter:

February 18th You must meet with your section leader to discuss ideas and brainstorm as necessary on or before this date. Your section leader will be able to help define the scope of the project to ensure it's advanced enough to require CS106X material, but not so advanced that you'll be at risk for not completing it.

February 25th You must send Jerry and your section leader a short email of no more than 300 words, and the content of this email should briefly describe your project, explain why you're interested in the problem you chose, and identify the CS106X material you'll rely on to build the final project.

March 4th You should meet with your section leader to ensure you're on track to complete your project without drama. You're welcome to reach out to the course staff as often as you need to prior to then, and you may redefine the project along the way, provided your section leader approves.

March 21st We'll convene as a group at 10:00am for a CS106X Final Project Poster Session, location TBD, where you'll present a research-style poster summarizing the problem you worked on, what was scientifically interesting about it, and what the primary implementation challenges were. You'll also want to be able to demo what you built and let other students play with it. You'll need to submit your entire implementation by 11:30am, once the poster session wraps up.