Chenyang Yuan

yuanchenyang@gmail.com
http://www.github.com/yuanchenyang

Education

Double Major in Computer Science and Physics Expected Graduation: 2016
The University of Berkeley at California, Berkeley, CA GPA: 3.94 (Technical: 4.00)

Technical Skills

Python ∼ 500 hours, used this language for both small tools and large projects

Java ∼ 100 hours, designed and implemented data structures for class projects

TeX ∼ 500 hours, used LaTeXto typeset documents since high school

Scheme ∼ 50 hours, written multiple interpreters for a subset of Scheme, have a good understanding of how it works

Javascript ∼ 50 hours, used Javascript to create a web-based Scheme interpreter

C ∼ 100 hours, wrote programs and compiled them for my graphing calculator

Assembly ∼ 50 hours, familiar with how CPUs work and how to program them

Emacs I do all my text editing with a highly customized version of Emacs

Selected Projects

Building a Computer from Scratch `https://github.com/yuanchenyang/My-EOCS`
Following the instructions from a book called the Elements of Computing Systems, I built a CPU from logic gates using a hardware simulator. Then I proceeded to create an assembler for the CPU and a VM simulator that takes in VM code (similar to java bytecode) and outputs assembly code.

Online SICP Textbook `http://xuanji.appspot.com/isicp/1-1-elements.html`
Made an interactive version of the classic Structure and Interpretation of Computer Programs book with my friend. I created the multithreaded Javascript-based Scheme interpreter.

Logic Gate Simulator `https://github.com/yuanchenyang/Logic-Simulator`
Used Python to create a logic gate simulation system with constraint passing. This system also allows powerful abstractions to be made so that more complicated sets of gates can be created, saved and reused. This project won an honorable mention in the Facebook Battle of the Bay hackathon.

Perfect Strategy for Hog `https://github.com/yuanchenyang/Hog-Perfect-Strategy`
For a project in my CS class, we have to create artificial intelligence agents to compete in a dice game called Hog. I used dynamic programming and recursion to create a prefect strategy that cannot be beaten, thereby winning the contest.

Work Experience

CS61A Reader, *UC Berkeley* *Spring 2013*

- Reader for the class Structure and Interpretation of Computer Programs
- Provided feedback and comments for students' code
- Held debugging sessions

Relevant Awards

First Place, Cal vs Stanford Big Hack *Apr 2013*
Honorable Mention, Facebook Battle of the Bay Hackathon 2012 *Oct 2012*
Rank 15, Hacker Rank Back to School Hackathon 2013 *Feb 2013*

Relevant Coursework

CS61A, Structure and Interpretation of Computer Programs *Fall 2012*
Introductory computed science class, ranked 3rd out of about 700 students.

CS61B, Data Structures and Algorithms *Spring 2013 (in progress)*

EECS70, Discrete Mathematics and Probability Theory *Spring 2013 (in progress)*