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: 3.967 (Technical: 4.00)

Technical Skills

Proficient in Python, Javascript, Emacs, HTML, jQuery, Java, LaTeX
Experience in C, Scheme, Haskell, SQL, Android, Assembly

Work Experience

Undergraduate Student Instructor for CS61A, *UC Berkeley* — *Fall 2013 – Present*

- I teach sections, prepare discussion notes and labs, help write the autograder for projects and hold office hours.

Software Engineering Intern, *Clover* — *July–August 2013*

- Helped improve internal tools
- Built an API auto-documentation system; designed and build an API Explorer: https://www.clover.com/api_explorer
- Created demo app using Clover's API: https://github.com/clover/example-server

Reader for CS61A, *UC Berkeley* — *Spring 2013*

- I provided feedback and comments for students' code and held debugging sessions

Selected Projects

Facebook Group Archiver https://github.com/yuanchenyang/facebook-group-archiver
A tool for saving Facebook groups in a local database and doing comprehensive searches locally. After the first download, it will sync the local database with the Facebook group during each run. Also includes a web-interface for stats, searching and doing database queries.

Interactive 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 asynchronous Javascript-based Scheme interpreter used on the website.

Scheme on TI-89 https://github.com/yuanchenyang/TI89-Scheme
Built a Scheme interpreter from scratch that runs on my TI-89 graphing calculator. It is written in C and supports a small subset of the Scheme language.

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.

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.

Relevant Awards

First Place, Cal vs Stanford Big Hack — *Apr 2013*
Created a scheme interpreter in C on my TI-89 graphing calculator
Third Place, Hackers at Berkeley HackJam — *Apr 2013*
Made an animation sequence on my TI-89 graphing calculator
Honorable Mention, Facebook Battle of the Bay Hackathon 2012 — *Oct 2012*
Build a logic gate simulator with a graphical interface in Python.
Rank 15, Hackerrank Back to School Hackathon 2013 — *Feb 2013*