Chenyang Yuan

yuanchenyang@gmail.com

http://www.github.com/yuanchenyang   http://www.chenyang.co

Education

Double Major in Computer Science and Physics
The University of Berkeley at California, Berkeley, CA

Expected Graduation: 2016
GPA: 3.977 (Technical: 4.00)

Technical Skills

Proficient in Python, Javascript, Emacs, jQuery, Java, LATEX
Experience in C, Scheme, Haskell, HTML, Hadoop, Android, SQL, 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   http://archiver.chenyang.co
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.

WebGL Particle Simulator   http://www.chenyang.co/particles
A simulation with thousands of particles attracted by gravity, created with WebGL and Javascript.

Python Control Flow Visualizer   http://pyvisualizer.chenyang.co
An online tool that run python programs and visualize the code branching using D3.js

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.

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 Nor-Cal Hackathon 2013   *Oct 2013*
Built a online Python code branching visualizer.
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*