Chenyang Yuan

yuanchenyang@gmail.com

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

Education

Double Major in Computer Science and Physics　　　　　　Expected Graduation: 2016
The University of Berkeley at California, Berkeley, CA　　　　　GPA: 3.939 (Technical: 4.00)

Technical Skills

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

Work Experience

Undergraduate Student Researcher, *UC Berkeley*　　　　　　*Spring 2014 – Present*
- I work with Professor Ras Bodik on the synthesis of a layout engine for an experimental browser, Servo. We specify CSS layout rules with an attribute grammar which compiles into a layout engine that takes in HTML and CSS and outputs the positions and sizes of every element. This layout engine, generated in Rust, then fits into the Servo browser.

Undergraduate Student Instructor for CS61A, *UC Berkeley*　　　　*Fall 2013 – Present*
- Teach sections and labs, holds office hours
- Help write the autograder for projects
- Wrote Javascript interpreters for Scheme and Logic languages used in the class, so that students can interpret code on their browsers without installing interpreters on their machines.
- Ran and maintained the codereview system used to give students composition feedback from readers

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

Math Competition Trainer, *National University of Singapore High School*　　*March 2012*
- Compile problems and create training notes
- Conduct classes for grade 8-10 students

Physics Competition Trainer, *National University of Singapore High School*　*March-August 2012*
- Prepare PhD-qualifying exam level problems
- Conduct classes for grade 11 students
- Create and grade a test

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　　　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.

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.

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*