SHANGWAY HSU

+1 408 368 9917
shangway.hsu@gmail.com
https://github.com/ShangwayHsu
https://www.linkedin.com/in/shangwayhsu

EDUCATION

University of California, San Diego — Expected Graduation: 2018
B.S. Computer Science, Minor - Cognitive Science — Cumulative GPA: 3.67 - Major GPA: 3.86

Related Coursework:

CSE 100: Advanced Data Structures and Object-Oriented Design
CSE 30: Computer Organization and Systems Programming
CSE 105: Theory of Computability
CSE 20: Discrete Mathematics
CSE 21: Mathematics for Algorithms and Systems
CSE 101: Design and Analysis of Algorithms

SKILLS

Proficient in: Java, C++, Python, HTML/CSS/JavaScript, Unix, Git
Working Knowledge: C, Swift/iOS/Xcode, SQL

EXPERIENCE

SLAC (Stanford Linear Accelerator Center) National Accelerator Laboratory — *Menlo Park, CA*
Intern — Jun 2015 - Aug 2015

- Implemented an optimization algorithm in C++ called Particle Swarm to find an optimal configuration for SLAC's LCLS (particle accelerator) in order to form a coherent electron beam.
- Used Python MATLAB library for graphical visualization of the optimization simulation data.
- Made improvements to the algorithm responsible for the electron beam bandwidth calculation.

De Anza Cupertino Aquatics — *Cupertino, CA*
Swim Instructor — Jun 2014 – Sep 2014

- Taught students ranging from ages 3 to 40 year olds using DACA's swimming curriculum.
- Learned how to communicate effectively with students and how to be creative in creating lesson plans.

PROJECTS

Personal Website – https://shangwayhsu.github.io — *Continuous*

- Implemented personal website to showcase projects through the use of BootStrap/HTML/CSS/JavaScript.
- Single-page website with scrolling animations and dynamic background.
- Responsive page with mobile support such as collapsible navigation bar to fit devices with smaller screen sizes.
- Email Contact form powered by Formspree.io.

To-do List App - iOS App Development — *Feb 2016*

- Implemented To-do list in Swift and Xcode for iPhone.
- Extensive use of StoryBoard in Xcode for UI elements and navigation control.
- Functionality includes adding new items with title and a short description, and deleting/editing existing fields.

Autocomplete - C++ — *Jan 2016*

- Used Multiway Trie to implement a dictionary capable of Autocomplete.
- Use of Priority Queue to store relations between nodes to decrease autocomplete time at the cost of space.
- Multiway Trie guarantees O(L) in find() and autoComplete(), where L is length of the longest word.

Unix Argument Parser - Java — *Apr 2015*

- Implemented a general command-line argument parser library following the Unix single-character option conventions and also the GNU long-form option conventions by using Object Oriented Design.
- Created using data structures such as Hashmaps and Sets for maintaining an organized OOP structure.
- Uses the fluent-interface style described in https://www.martinfowler.com/bliki/FluentInterface.html.

Replica of 2048 Puzzle - Java — *Mar 2015*

- Backend created using Java using matrices to represent the tiles of the game.
- GUI created using JAVAFX 8 and supports window resizing.
- Additional functionality includes: saving game states, loading games from text file, and multiple game grid sizes.