Edmund Loo

405 E. Colorado Boulevard, Arcadia, CA 91006
(626) 423-9065 • looe@uci.edu
https://www.linkedin.com/in/edmundloo

EDUCATION

Bachelor of Science in Computer Science and Engineering **Expected December 2016**
University of California, Irvine Dean's Honor List, **GPA: 3.36**

TECHNICAL QUALITIES

- Fluent in C++ and Python; Have worked with Java, C, C#, OpenGL, Qt Libraries, Arduino, Assembly, VHDL, TACC
- Experience working with Linux, command line, Perforce, Git, debuggers, IDEs, and a variety of other development tools
- Knowledge in algorithms, data structures, object and attribute oriented programming, networking, and operating systems
- Coursework in software engineering methods, signal processing, discrete math, and logic based design
- Strong enthusiasm and capabilities in adapting, learning, problem solving, research, and software development

CAREER HISTORY

Arista Networks, ***Software Engineering Intern***, *Santa Clara Headquarters* **June 2015 – September 2015**
- Developing software for multi-chassis link aggregation (MLAG) on Arista's Linux-based Extensible Operating System
- Optimized a commonly called MLAG failure recovery process to be significantly faster for shipping on future Arista switches
- Adapted to Arista's development environment which uses Perforce, a full Linux command line interface, and many Arista tools

Computer Learning Center, ***Computer Science Tutor***, *Pasadena City College* **January 2014 – August 2014**
- Pioneered the Computer Science Tutoring Program as a facilitator and a first generation tutor
- Amplified progress and improvement in the Computer Science Department by collaborating with faculty and other students through localized educational research and peer surveying
- Generated success and retention in computer science courses, improvement was seen in over 90% of students and drop rates decreased from an average of 40% in some classes to under 10%
- Mentored over 80 students throughout 3 lower division Computer Science courses

PROJECTS

Game of Life, ***Python Inheritance and Simulation*** **May 2015 – June 2015**
- Cooperated with another classmate to develop an inheritance-heavy graphical prey and predator ecosystem simulation
- Fabricated an API with a large inheritance hierarchy for building similar simulations that allows the placement and graphical display of objects that will graphically exhibit the pre-programmed behavior attached to each of them
- Achieved one of the highest scores in the course, demonstrating advanced knowledge in Python and inheritance

California Plug Load Research Center, ***Back-End Networking on Embedded Systems*** **October 2014 – April 2015**
- Government and privately sponsored energy efficiency research initiative with focus in micro-grid power and plug load systems
- Assembled functions and controls in C++ for an Arduino and an Adafruit Wi-Fi shield that allowed the Arduino to find and connect to a Wi-Fi network and continuously send several bytes of data to a local Python server through a TCP socket
- Built a local networking server in Python that constantly listens for data from a TCP socket connected to the Arduino, parses and creates a JSON object from the data, and sends the JSON object to both a local storage file and a web-hosted MongoDB

UAV Forge Networking Team, ***TCP Networking in a Graph*** **October 2014 – January 2015**
- Team-based, ongoing, research and design project to fabricate a military specification unmanned aerial vehicle from scratch
- Conducted the C++ programming members within the networking team by researching for and directing the members in the right direction, discussing specifications with team leads, and translating required specifications into implementation details
- Designed a mesh network simulation using a graph structure where edges were implemented as TCP sockets and nodes as I/O devices, the implementation included a routing table and the ability for the routing table to regenerate itself if a node goes down

The Editor's Dream, ***Text Parser Using Heap and Binary Tree*** **October 2013 – November 2013**
- Created a program that parses a block of text, orders and counts individual words, then graphically displays results under strict runtime requirements; this was implemented twice using both an orchard of self-balancing binary trees and heaps
- Implemented data structures and their corresponding standard algorithms with vector of pointers and pointer implementations demonstrating thorough knowledge in C++ and pointer programming
- Completed the parsing and word recognition algorithms, the information processing, and the graphical user interface using Qt libraries, topping the class grade sheet in the process due to elegance and performance of implementation