Jun Hyung Lee

808-557-0383 | junhyunglee.jl@gmail.com | linkedin.com/in/jun-hyung-lee | junhyunglee.com

Education

University of California, Berkeley — Berkeley, CA
Bachelor of Arts in Computer Science, Minor in Education — *Aug. 2020 – May 2023*
GPA: 3.83

Technical Skills

Languages: Java, Python, C++, C#, LaTeX, SQL, HTML, CSS, JavaScript
Frameworks/Tools: Git, Unity, Node.js, Flask, JUnit

Experience

Data Management Associate — Mar. 2022 – Present
Haas School of Business — *Berkeley, CA*
- Developing application that compiles company funding information from databases like Pitchfork. Estimated to cut data cleaning process by roughly 50%.
- Perform statistical analysis on information stored in datasets
- Perform routine data cleaning to maintain integrity of databases

Information Technology Intern — Nov. 2021 – Jan. 2022
Huntington Ingalls Industries — *Honolulu, HI*
- Support the IT team in maintaining hardware, software, and other systems in areas such as cybersecurity, programming, analytics, and data center management
- Perform on-site installation and preventative maintenance on PCs and related peripherals
- Ensure that hardware and software systems are deployed, implemented, and functioning

Programming Tutor — Aug. 2021 – Present
Breakout Mentors — *Palo Alto, CA*
- Skills Taught: C++, C#, Java, Python
- Data Structures Taught: Singly-linked and doubly-linked lists, binary search trees, hashmaps, vectors, multi-dimensional arrays
- Core Concepts Taught: Recursion, object-oriented programming, tree traversal, algorithm time complexity
- Created lesson plans designed to teach grade-school students concepts relating to data structures and game development

STEM Camp Programming Instructor — June 2018
Kealakehe High School — *Kailua-Kona, HI*
- Languages Taught: C++, Scratch, AppInventor
- Taught a class of students how to program own desktop and mobile applications

Projects

Gitlet | *Java*
- Developed a Git-like offline version control system for text files
- Functionality includes add, commit, remove, branch, reset, merge, and merge conflict-detection

Procedural Maze Generator For Rogue-like Game | *Java*
- Designed and programmed procedural maze and hallway generator for a top-down 2D roguelike game, along with the game's UI, collision detection, and audio systems

Scheme Interpreter | *Python*
- Developed a lisp-like interpreter in Python
- Required an intricate understanding of syntactical and lexical analysis to implement

Running Penguin Game | *C#, Unity*
- Programmed and modeled assets for endless-runner game patterned after Subway Surfers in C#

Retro Text-Based Role-Playing Game | *C++*
- Programmed text-based game featuring robust character customization options with over 10 different playable classes and races, along with over 400 explorable locations