

WINSTON TANG

Berkeley, CA

☎ +1(626)321-0675 ✉ winston0753@berkeley.edu [in linkedin.com/in/winston0753/](https://www.linkedin.com/in/winston0753/) github.com/winston0753

EDUCATION

University of California, Berkeley

Bachelor of Arts in Data Science and Economics

Expected May 2024

Berkeley, CA

RELEVANT COURSEWORK

- Designing Information Devices and Systems
- Data Structures
- Structure and Interpretation of Computer Programs
- Linear Algebra
- Foundations of Data Science
- Multivariable Calculus

EXPERIENCE

Mathematics Tutor | Support for All

Aug 2018-May 2019

- Tutored primarily elementary to middle school students of Algebra I levels and higher
- Guidance in homework issues regarding arithmetic and basic trigonometric concepts
- Introduced relevance of mathematics through interacting with physical tools and hands-on activities
- Qualified for Presidential Gold Service Award with 100+ hours of service

PROJECTS

Gitlet | Java

2021

- Implemented a simpler version of version control system Git
- Implemented basic functions capable of allowing users to commit, branch, and merge files based on input commands
- Used hash functions to track file versions, and stored file data in arrays and hashmaps across branches and commits

Scheme | Python

2020

- Developed an interpreter for a subset of the Scheme programming language
- Implemented syntactic analysis to recursively parse Scheme tokens into interpreter's Python representation of Scheme expressions
- Created evaluator to be able to call primitive procedures and special forms to be applicable on its corresponding value in Python

Enigma | Java

2021

- Created a simulator version of German Enigma encryption machines used during WWII
- Implemented a progressive substitution, where each subsequent character of the message provided a different permutation, as opposed to static substitution ciphers
- Used array functions to store and progress each permutation pattern with each subsequent character of an inputted message

Jump | Java

2021

- Implemented a simulator based on the two player board game Lines of Action
- Used min and max game trees to create AI computer to find the move with highest heuristic value
- Created GUI to display board, interpret mouse inputs, and ability to switch between player and AI

TECHNICAL SKILLS

Programming Languages: Python, Java, JavaScript, C++, CSS/HTML, R, Scheme

Developer Tools: VSCode, BlueJ, IntelliJ

Languages: English (native), Chinese/Mandarin (native), Spanish