

JESSICA YU

jessica.yu@berkeley.edu ◇ (510) 926-7016 ◇ Fremont, CA ◇ linkedin.com/in/jessicayu00

EDUCATION

University of California, Berkeley

May 2022

B.A. Computer Science

Coursework: *CS162: Operating Systems, *CS188: Artificial Intelligence, CS61C: Computer Architecture and Machine Structures, CS170: Algorithms, CS70: Discrete Math and Probability Theory, CS61B: Data Structures, CS61A: Structure and Interpretation of Computer Programs

EXPERIENCE

Hearst Museum of Anthropology Undergraduate Researcher

Sep 2019 to Dec 2019

- Digitized UC Berkeley's collection of archaeological artifacts through photogrammetry.
- Used Metashape and Blender to process and refine models.

TEACHING

Computer Science Mentors (CSM) at UC Berkeley

Sep 2019 to present

- Lead weekly small-group mentoring sections of 4-5 students on core topics of the course.
- CSM is the largest student-run volunteer teaching organization on campus.

UC Berkeley Computer Science Department Course Staff

Jun 2019 to present

Course Tutor:

- Lead small group tutoring sections of 4-5 students on core course material.
- Coordinate with TAs to grade projects and exams, staff office hours, and respond to questions on the class forum.

Teaching by Course:

CS88 (Computational Structures in Data Science): 1 semester CSM

CS61A: 2 semesters CSM*, 2 semesters Course Tutor

CS61C: 1 semester CSM, 1 semester Course Tutor*

PROJECTS

Enigma [CS61B]

Spring 2020

- Performs all computational aspects of the Enigma machine used by the German military during WWII.
- Takes in textual input and returns encoded / decoded text.
- Approx. 1000 lines in Java.

Lines of Action [CS61B]

Spring 2020

- Simulates the checkerboard board game Lines of Action using terminal commands and GUI input
- Approx. 1000 lines in Java

Mandelbrot [CS61C]

Fall 2020

- Generates a visual representation of the Mandelbrot function. Worked with memory and file I/O
- Approx. 500 lines in C

CPU Logic Design [CS61C]

Fall 2020

- A two-stage pipelined datapath from scratch in Logisim that is capable of running the RISC-V instruction set
- Designed a system of controls to differentiate behavior of CPU based on distinct instructions
- Streamlined visual and logical components and accounted for control hazards

Languages: C, Java, Python, Scheme, RISC-V

* = *in progress*