Jamie Ip

(650) 787-1828 | jamieip@berkeley.edu | jamieip.com

Education

UC Berkeley *Berkeley, CA*

Bachelor of Arts in Computer Science, GPA: 4.0/4.0 *Expected Graduation: May 2022*

- **Honors:** Member, Upsilon Pi Epsilon
- **Coursework:** Artificial Intelligence, Data Structures, Efficient Algorithms, Computer Architecture, Foundations of Data Science, Ethics of Data, Teaching Computer Science

Skills

Languages Python, Java, C, Scheme, LaTeX

Graphic Design Photoshop, Gimp, Procreate

Experience

Berkeley Institute for Data Science *Berkeley, CA*

Research Intern; Mentor: Dr. R. Stuart Geiger *September 2019 - PRESENT*

- Investigating machine learning classification papers, gathering and analyzing information on the reliability of the training data used in published machine learning research
- Coded programs in Google Apps Script to assign labels in Google Sheets and create visuals from data

GamesCrafters Research and Development Group *Berkeley, CA*

Research Intern; Mentor: Dr. Dan Garcia *January 2020 - PRESENT*

- Designed strong solvers in Python for complete information two player strategy games
- Created a Unity interface for solving and playing 3x3x2 Tic-Tac-Toe, capable of solving positions instantaneously using hashing and file indexing, 30 minutes faster than solving using memoization and symmetry removal

Computer Science Mentors *Berkeley, CA*

Senior Mentor *September 2019 - PRESENT*

- Providing weekly lessons, guidance, and resources to a small group of students in Berkeley's introductory computer science class, CS 61A, teaching Python, Scheme, and SQL
- Rated an average of 4.71/5 in overall teaching effectiveness and 4.86/5 in helpfulness

Projects

Phlow

UC Berkeley Bear Jams Fall 2019, 1st Place

- Worked with four others to design a 2D endless runner snake game with original art and music
- Written in C#, designed in Unity

RageCage

Cal Hacks 5.0

- Collaboratively designed a Google Chrome browser extension that monitors writing tone on social media, emails, and online forums to warn its user upon detecting excessive amounts of anger
- Written in HTML and CSS, utilizes the IBM Watson API for detecting and quantifying emotions

Reinforcement Learning

- Implemented backend code for value iteration and Q-learning algorithms used by self-learning Crawler and Pacman AI agents, simulating the world as an unknown Markov Decision Process
- Written in Python, using no external AI libraries

Build Your Own World

- Designed an engine that randomly generates 2D tile-based worlds with distinct rooms, hallways, enemies, and a player avatar that can interact with the world
- Recursively partitions the world into randomly sized cells to generate unique rooms, then connect them with hallways
- Written in Java