# Owen **Jow**

*2530 Hillegass Avenue #302, Berkeley CA 94704*

(650) 288-8553 | owenjow@berkeley.edu | owenjow.xyz | ohjay | owenjow

## Education

**University of California, Berkeley** — *Berkeley, CA*

Bachelor's in Computer Science — *August 2014 - PRESENT*

- Relevant courses: Deep Reinforcement Learning*, Special Topics in Deep Learning*, Optimization Models in Engineering*, Computer Graphics, Advanced Computer Graphics, Algorithms, Machine Learning, Image Manipulation & Computational Photography, Artificial Intelligence, Ruby on Rails, Operating Systems, Database Systems, 3D Modeling & Animation, Data Structures, Linear Algebra
- GPA: 3.8 (some form of A in all completed courses listed above)

* *in progress*

## Experience

**Robot Learning Lab (Prof. Pieter Abbeel)** — *Berkeley, CA*

Undergraduate Researcher — *June 2016 - PRESENT*

- Developed a system for complex, autonomous robot control by means of a neural network trained to imitate VR demonstrations.
- Worked on the platform for learning from demonstration (LfD) as well as the VR interface for robot teleoperation.
- Publications: *Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation* (submitted to ICRA '18)

**UC Berkeley EECS Department** — *Berkeley, CA*

Undergraduate Student Instructor — *August 2015 - PRESENT*

- Served as a teaching assistant for CS 61A, the introductory programming paradigms class at Berkeley (~1600 students in Fall 2016), and CS 194-26, the computational photography class.
- Worked with professors and other TAs to create course content such as exams and section worksheets.
- Led weekly discussions, labs, and office hours.

## Highlighted Projects

**Deep Blue and Gold | Python, TensorFlow** — *April 2017 - PRESENT*

- Chess engine capable of either emulating a certain style of play (using an evaluator network trained with supervision data) or playing as optimally as possible (using an evaluator network trained with temporal difference reinforcement learning).
- For move selection, uses minimax with various extensions (e.g. quiescent search and probability-based termination).

**Single View Modeling | Python, OpenGL** — *December 2016*

- PyOpenGL realization of the "tour into the picture" algorithm, complete with a GUI for selecting points and walking through scenes.
- In its current incarnation, the program is able to take in a one-point perspective image and reconstruct a 3D model of its content.

**Automatic Image Stitching | Python** — *November 2016*

- Automatically and seamlessly stitches images together using Harris corner detection, feature descriptor matching with SSD, RANSAC, and homography estimation for warping.
- Also allows for the rectification of image features and the creation of 360° cylindrical panoramas.

**Lens Simulator | C++** — *March 2016 - April 2016*

- Path tracing, where rays are refracted according to an input lens model. Supports contrast-based autofocus.
- At its core: a physically-based renderer built upon a large number of ray intersection tests, acceleration using a BVH, Russian roulette methods for secondary ray termination, and reflection/refraction computation for different materials.

## Skills

### Languages, Libraries, and Frameworks

**Python**, C++, TensorFlow, PyTorch, Java, OpenGL, JavaScript, CSS, C, SQL, ROS, C#, Unity, Android, Ruby, Ruby on Rails, Django, Bash

### Software

Adobe Photoshop, Adobe Animate CC, Autodesk Maya