# Jeffrey Li

2521 Piedmont Avenue
Berkeley, CA, 94704

Email: lijw@berkeley.edu
Cell: 336-554-6880

## Education

**University of California, Berkeley**, Berkeley, CA **2015-2018 (expected)**

B.A. Computer Science, GPA: 3.80/4.00

## Academic Honors

**National Merit Finalist** **2015**

Scored in the top 1% of PSAT test takers.

**AP Scholar with Distinction** **May 2013**

## Projects

**Tennis at Cal Website** **Fall 2016**

I wrote the code for and designed the Tennis at Cal website using HTML, CSS, and jQuery. The website provides information on practices, tournaments, registration, and a biography for each of the officers. I implemented a navigation button that shows all options to navigate to by shifting the website when clicked on, as well as a carousel of pictures of people involved in the club from past semesters.

**Drum Machine** **Summer 2016-Present**

Using byte-arrays, JavaFX, linked-lists, and WAV files, I created a drum-machine that the user that use to create beats composing of open and closed-hihats, snares, kicks, and claps that are played at different tempos. The user can select which sound he wants played 16 different beats. By fusing the byte-arrays of each WAV file instead of using threads, the user can play the sounds without causing delay or lag. I plan to add functionality to allows the user to upload his own sounds to accompany the drum-machine.

**BearMaps** **Spring 2016**

Using a quad-tree that contained pictures of the city of Berkeley and the A* search algorithm, I created a working shortest-path finder between any two locations in Berkeley. Each building/important landmark on the map served as a node, and each street served as a link between landmarks. Also, by using a trie, I implemented an auto-search for the map that allowed the user to look up the names of the landmarks on the map.

**Text Editor** **Spring 2016**

Using JavaFX and a doubly-linked list, I created a working text editor that allowed the user to type, delete, start a new line, and use the mouse and cursor to reposition the typing cursor. I also implemented open and save functions to allow the user to save and re-open his work, as well as window-resizing.

**Scheme Interpreter** **Fall 2015**

Using Python, my partner, Andrew Martinez, and I wrote an interpreter that is able to read Scheme code. Scheme is a language that revolves around lists and pairs, so by utilizing classes that represent a pair and the "nil" object in Scheme, I created syntax reader and evaluator functions that allowed the user to type in Scheme.

## Skills

**Programming:** Python, SQL, Java, HTML, CSS

## Courses

CS61A (Structure and Interpretation of Computer Programs), Math 54 (Linear Algebra), CS61B (Data Structures), CS70 (Discrete Math and Probability Theory), Multivariable Calculus

## Interests

Tennis, College Basketball, Chance the Rapper.