Viraj Mahesh

virajmahesh@berkeley.edu | +1 510-646-5944

EDUCATION

UNIVERSITY OF CALIFORNIA BERKELEY

Expected: May 2017
B.S in Electrical Engineering and Computer Science
Dean's List (All Semesters)
Cum. GPA: 4.0 / 4.0

DELHI PRIVATE SCHOOL DUBAI

SAT I: 2290 SAT II: 800 PHY / 800 MATH II / 800 CHEM

LINKS

Github: github.com/virajmahesh
LinkedIn: linkedin.com/in/virajmahesh

COURSEWORK

UC BERKLEY: Structure and Interpretations of Computer Programs **(CS61A - Python)** • Data Structures **(CS61B - Java)** • Calculus **(MATH 1A & 1B)** • Mechanics and Wave Motion **(PHYSICS 7A)**
Intended: Machine Structures **(CS61C)**, Multivariable Calculus **(MATH 53)**, Linear Algebra and Differential Equations **(MATH 54)**
COURSERA: Principles of Microeconomics - UPenn, Machine Learning - Stanford

SKILLS

Proficient
Java • C++ • Python • JavaScript
Strongly Familiar
MATLAB • R • Android • CSS • HTML
Prior Experience
PHP • SQL • Git • Shell

AWARDS

Dean's Honor List (Fa 13, Sp 14): Requirements include completing 12 or more letter-graded units in that semester; a semester and overall GPA in the top 10% of all College of Engineering undergraduates.
Kraft Award for Freshmen: Attain a 4.0 GPA at the end of your first (fall) semester at UC Berkeley. Honors to Date

EXPERIENCE

SOFTWARE ENGINEERING INTERN | INFOSYS

June 2014 – August 2014 | Bangalore, India
Created a cross-platform UI Automation framework using Java, Jython and OpenCV. Researched and implemented 4 different kinds of image recognition techniques. Built a diagnosis automation framework that uses belief networks to model systems and calculate value of information. The framework is capable of creating a belief network from historical observation and fault data. All code was perfected, reviewed and merged with the existing code base.

UNDERGRADUATE RESEARCHER | BERKELEY LABORATORY FOR AUTOMATION

January 2014 – May 2014 | Berkeley, CA
Research Mentor: Professor Kenneth Goldberg
Worked on improving Eigentaste, a constant time collaborative filtering algorithm. Used greedy algorithms to make optimizations to Eigentaste. Learnt data analysis techniques such as principal component analysis and k-means clustering. Showed that using k-means clustering in Eigentaste leads to a 6% reduction in error rate.

SNAP DEVELOPMENT | EXTENSIONS DEVELOPER

February 2014 – Present | Berkeley, CA
Developed JavaScript extensions for Snap, a graphical programming language developed at UC Berkeley. Developed a GUI extension that allows users to take screenshots and export them as PNG files. Extended Snap in order to automatically rename costumes to prevent naming conflicts. Used Git and GitHub in order to track project changes and integrate changes into the main project.

CS61A COURSE STAFF | LAB ASSISTANT

February 2014 – May 2014 | Berkeley, CA
Assisted students with projects, lab work and homework. Helped explain key concepts such as recursion and operations on data structures such as traversal, reversal, merging, etc. Helped students understand debugging techniques. Explained good coding practices and code organization techniques.

PERSONAL PROJECTS

PERSONAL WEBSITE | 2014

Created a personal portfolio using HTML, CSS and JavaScript + jQuery. Created the design from scratch, without the use of any frontend libraries. Used jQuery and JavaScript to add UI effects such as animation and transition. Created a contact form to allow visitors to email me without revealing my email address using a backend written in node.js and hosted on heroku.

FALLING BALLS | 2013

2D Android game built using the Android SDK. Implemented Java libraries to handle game physics, collision detection and response. Used the AdMob SDK to display in-game advertisements. Used an SQLite Database to store high scores locally.

3D POOL | 2013

Created a 3D pool simulation table using OpenGL and C++. Added sound and 3D texturing to make the game more realistic. Implemented C++ libraries to handle rigid body simulation, collision detection and impulse based collision response. Used the glut wrapper for OpenGL to render 3D objects with lighting effects to improve realism.

2D BASKETBALL | 2012

Created a 2D Basketball simulation using OpenGL and C++. Added sound and texturing to make the game more realistic. Implemented C++ libraries to handle physics simulation, collision detection and event handling. Used PHP and MySQL to create an online database to handle player login, registration and high score submission.