Safi Mohammed

5104496885♦ safi.m@berkeley.edu ♦ github.com/grep-safi ♦US Citizen

Education

University of California, Berkeley | **Berkeley, CA**

B.S. Electrical Engineering & Computer Science (EECS) | *June 2019 - May 2022*

GPA: 4.0/4.0

Coursework: Data Structures, Computer Architecture, Structure of Computer Programs, Designing Information Devices and Systems I/II, Discrete Math & Probability Theory

Foothill College | September *2018 - June 2019*

A.S Computer Science

Work Experience

Lawrence Berkeley National Laboratory | **Berkeley, CA**

Undergraduate Researcher | *June 2020 – December 2020*

- Researched methods of parsing and visualizing cluster data in the Cori supercomputer's 12,056 compute nodes.
- Developed a novel visualization technique using a tree-map and heat-map hierarchical structure.
- Incorporated time series of CPU load and memory usage for every compute node over 24 hours.
- Received return offer for Fall 2020 to integrate my program using React.js and D3 in LBNL's NERSC user-page.

Projects

Physics Simulations

Developed a collection of responsive, web-based astronomy simulations using React, PIXI, and D3.

- Cosmological Redshift Simulator models the path and redshift of a 400 nm photon in an expanding universe.
- Hydrogen Atom Simulator demonstrates mechanics of electron excitation, de-excitation, and ionization.
- Planetary Configurations Simulator models the movement of planets around the sun in the Copernican system.
- Doppler Shift Simulator illustrates the observed wave frequency given moving source and target objects.
- More physics and astronomy simulations can be found on my Github.

2D Tile Game

- Implemented from scratch a pseudorandom world generator that creates distinct maps given only a seed number.
- Each world contains 2D rooms, corridors, enemy NPCs, torches, and the ability to interact with game elements.
- NPC characters were programmed with the A* pathfinding algorithm to locate and chase the player.
- Game mechanics included ability to shoot limited fireballs and find the key to a locked door to win the game.

Chess AI

- Built the game of Chess completely from Scratch using Python and the Tkinter GUI library.
- Combined point-value position boards and piece-value material scores to evaluate player's overall strength.
- Created an AI that used the Minimax algorithm and Alpha-Beta Pruning optimization to efficiently generate decision trees.
- Heuristic for determining best move used a weighted combination of material and position scores.

Scheme Interpreter

- Using Python, built a REPL interpreter for Scheme which included evaluation, lambda functions, and special forms
- The read functions parse Scheme tokens into Python representation of Scheme expressions
- The eval/apply functions handle special forms and user-defined procedures by creating and evaluating call frames.
- Print function returns the string representation of the output to the interpreter for printing.
- The loop function continues the cycle while retaining relevant local variables and function names.

Skills/Tools

Languages: Python (proficient), Java (proficient), Javascript (proficient), C++, Scheme

Node.js, D3.js, PIXI.js, Webpack

Web Development: React.js, HTML, CSS,