Eric Feng

240-481-1269 • ericfeng@terpmail.umd.edu • https://ericfeng52.github.io/portfolio/

EDUCATION

US Citizen | University of Maryland, College Park | Bachelor of Science in Computer Science Expected May 2024

Relevant Coursework

- Object-oriented programming, Algorithms, Web Application Development with JavaScript, Introduction to Computer Systems, Discrete Structures, Organization of Programming Languages, Data Structures, Introduction to Data Science, Design and Implementation of Programming Languages, Compilers Linear Algebra, Calculus I/II

SKILLS/INTERESTS

Languages: Java, Python, C/C++, JavaScript (&TypeScript), Ruby, OCaml, Rust, HTML, CSS, JSON, Ajax, Swift, x86, Racket
Frameworks/Libraries: React, JUnit, Jupyter, Node.js, MongoDB, Pandas/NumPy, Spring
Tools: Git, Linux, Docker, APIs Algorithm Analysis, Jira, Agile, Scrum, GitHub

EXPERIENCE

Ameritas | Lincoln NE

Package Systems Developer Intern (Remote) *May 2023-Aug 2023*

- Resolved bugs within a newly implemented healthcare application, cutting the amount of **JIRA** issues from **150 to 30** throughout the course of the internship
- Debugged application **codebases** using **Firelight APIs** to ensure precise data translation
- Conducted **User Acceptance Testing (UAT)** by simulating real-world scenarios to ensure application effectiveness
- Worked in an **Agile** environment and maintained transparent communication with other teams such as **QA**

National Institute of Health | Bethesda MD

Mobile Development Intern *April 2017-Sep 2017*

- Developed a 2D educational game for iOS devices using **Swift** which taught users about environmental health and toxicology
- Worked within a 3-person team to implement the physics (sprite collisions and interactions) and graphics of the game

PERSONAL PROJECTS

RASM, Racket + x86 2023

- Implemented a **compiler** which translated **Racket** code into **x86 Assembly** code
- Defined the **AST (Abstract Syntax Tree)** type and **structure definitions** for numeric and boolean operations, and forms of control flow expressions
- Developed a recursive **parser** that tokenized inputs that were later used in the **interpreter** and compiler programs

The Effect of Per Game Stats on the Modern Era NBA Teams Win Percentage, Python 2023

- Created a tutorial that walked users through the entire **data science pipeline**: data curation, parsing, and management; exploratory data analysis; hypothesis testing
- Pulled NBA data from an online **database**, used tools such as **NumPy**, **pandas**, and **beautifulsoup** to **model**, organize, analyze, and draw hypotheses with the data

Personal Portfolio, HTML + JavaScript + CSS 2023

- Developed a responsive personal website utilizing **HTML**, **JavaScript**, and **CSS** to showcase my software engineering projects, technical skills, and professional experience.

Unix Simulator, C 2022

- Simulated Unix system in **C** where there existed a root directory, additional directories and files which users could add/modify, and functions such as getting the path from the current directory to root
- Written using **dynamically allocated data structures**, implemented concepts such as **memory allocation** and **memory deallocation**, and searching algorithms such as **DFS** and **BFS**

Six Degrees of Kevin Bacon, Java 2022

- Simulated a web of profiles on social media using **graphs** in **Java**
- Performed operations such as finding the least amount of connections between two profiles (implemented using **DFS**) and finding all first connections (implemented **BFS**)

MicroOcaml Interpreter, OCaml 2022

- Created a version of **utop** (a top-level for OCaml) which read user inputs as commands and executed them
- Constructed **lexer** and **parser** to tokenize and check the command grammar, then implemented an **interpreter**