Alek Schmierer
alek.schmierer@.com
alekschmierer.github.io
209-327-1110

Education

Bachelor of Science in Computer Science May 2025
Arizona State University Tempe, Arizona
Cumulative GPA 3.83

Experience

Software Engineering Capstone Intern, Living In Silico August 2024 – Current Remote, Az

- Contributed to the design and full-stack development of a software application integrating external tools for immunomic data analysis.
- Implemented functionality using Python, Conda, HTML, CSS and JavaScript ensuring seamless integration for an interactive user interface.
- Collaborated with a team to process and visualize immunomic data for receptor-ligand binding studies.

Coding Projects

Fine Motor Skills [JavaScript, Google AI][2022]

- Designed and implemented a fine motor skills game with JavaScript that challenges children to identify shapes while writing neatly
- Used Google's Teachable AI machine learning library to create a custom model that analyzes the players camera and allows them to interact with the game.

Blocky's World[GDScript, C++, C#][2024]

- Developing in Godot and is a rouge-like game created as a personal project
- Defined a design document to prove my ability to plan, organize, and communicate complex game concepts.
- Demonstrated proficiency in game development, C++, data structures, and problem-solving

Compiler for Intermediate Representation [C++] [2024]

- Designed and implemented a compiler to convert input programs into a linked list
- Created instruction nodes specifying types, operands, and execution flow.
- Utilized provided content (execution framework, lexical analyzer, and input buffer) to focus on developing the instruction list and traversal logic

EffortLogger [Java, SQL, FXML][2023]

- Developed an app that tracks effort of employees and analyzes story points to track the amount of effort processes take for the finished product
- Worked in a team of three people to produce a finished product that also included documentation, testing, and debugging

Dijkstra's Algorithm [C++, C][2023]

- Implemented Dijkstra's algorithm in C++ and C to solve shortest path problems in weighted graphs.
- Utilized data structures such as min heap, stack, and adjacency list to efficiently manage graph data
- Successfully found the shortest path between any two valid points in the graph, demonstrating my understanding of optimization of algorithms.

Abstract Syntax Tree Parser and Type Checker [C++] [2024]

- Implemented a parser and precedence table to process a custom language, and created an abstract syntax tree
- Utilized provided content (execution framework, lexical analyzer, and input buffer) to focus on designing core parsing logic, language rules and ensured correctness with type checking.
- Enhanced understanding of compiler design principles, including parsing, syntax analysis, and semantic validation