

Dylan Oldham

(541) 321-9889 | dylansoldham@gmail.com

Portfolio: dylansoldham.github.io/portfolio

EDUCATION

Bachelor's Degree in Computer Science

June 2024

Oregon State University, Corvallis, OR
GPA 3.89

EXPERIENCE

Software Engineer Intern

Jun. 2022 - Sep. 2022

ReplayLogic, Remote

- Developed the early stages of a full stack software product which detects potential IAM misconfigurations and cloud system vulnerabilities.
- Prototyped new UI elements for allowing users to intuitively manipulate complex, recursively defined data on a webpage.
- Created new backend services in Rust and Java utilizing REST endpoints and PostgreSQL.
- Designed unit test suites for backend services and created CI workflows on GitHub to run them.

Service Desk Technician

Dec. 2021 - May 2022

Oregon State University, Corvallis OR

- Diagnosed and addressed the technical issues of university faculty and students over the phone.

Software Engineer Intern

May 2020 - Jun. 2022

Group6 Studios, Corvallis OR

- Designed, implemented, and tested gameplay features for an online multiplayer game.
- Optimized the game render loop to improve its average frame rate by over 400%.
- Implemented new features in the low-level graphics, UI/UX, animation, game logic, and network communication.
- Collaborated with artists and designers to translate their ideas into functional code.

SKILLS

C, C++14, JavaScript/TypeScript, Java, Rust, Haskell, Python
React, OpenGL, GLSL, CUDA, Sqlite, PostgreSQL
Doxygen, GoogleTest, CI/CD, GitHub Workflows
Agile Development

PROJECTS

GPU-Based Modeling of Material Degradation Inside of a Nuclear Reactor

- This was my senior capstone project at OSU. The project goal was to take a mathematical model of how radiation damage evolves in a crystalline material and convert it into code which could utilize the GPU for acceleration.
- Collaborated 12 other people, including 11 computer science students and a material scientist advisor, to formulate specific requirements and draft
- Wrote the core simulation, which involved reading models described in papers, finding a numerical method which was appropriate for solving the model, and breaking that method into pieces that could be run in parallel using CUDA.