BRADLEY ALDRIDGE

Software Developer

roachcode@gmail.com
(778) 676 - 3030
Victoria, B.C.
github.com/roachcode

SKILLS

C++ (OOP)

JS, TS, Node.js
Cypress (briefly)

Creative Problem-Solving
Personal Accountability / Humility
Communication / Active Listening
Interpersonal Skills / Diplomacy
Emotional Intelligence / Sensitivity
Determination / Desire To Learn
Passionate Work Ethic
Adaptability / Flexibility
Perfectionism vs. Pragmatism

Check out my website for
Some examples of my work:

www.bradleyaldridge.com

CAREER OBJECTIVE

Inspired and enthusiastic self-taught developer looking for a long-term role where I can grow with a team and continue to learn from talented and experienced Developers. I'm a quick study and I like to make things - correctly.

EMPLOYMENT

Junior Developer / Quality Assurance Analyst

Developer, Agile **North Orca Technologies**

May 2022 – Feb 2023

- Developed front- and back-end software in a Dockerized environment
- Reliably met or exceeded goals for quality and robustness
- Delivered beyond MVP within alotted timeframes when feasible
- Followed RESTful API best practices while managing complexities beyond CRUD
- Maintained positive work relationships through sincerity and clear communication

Asst. Manager, Hardwood Installer, Clerk, Cook, Labourer

Adaptive Employee **Various Places**

May 2022 – Feb 2023

- Various blue-collar jobs interacting with people from all walks of life. From this experience I transfer people skills, work ethic, and strong problem solving skills.

PROJECTS

Ray Tracer

Developer (Solo)

April 2022 - ongoing

- Low level image processing, implementing custom reflections and refractions over a number of spheres.
- Required some math – implementing quadratic formulas and linear algebra.
- This raytraced image renders slow as the math is done largely on the stack. This is changing as I port the ray tracer to SFML and translate the processing to GLSL.

2D Graphics and Game Engine

Developer (Solo)

April 2021 - ongoing

- Utilized C++ with the SFML library (OpenGL) to create a 90's era JRPG game
- Created an efficient rendering pipeline that minimizes draw calls using tilemaps and off-screen render targets
- Implemented a vector flow grid and particle emitter to simulate fluid dynamics
- created a 4D toroidal algorithm to generate tileable, fractal patterns from OpenSimplexNoise (Perlin noise)
- Created responsive custom windows and menus from simple primitives
- Intercepted keyboard strokes without exposing vulnerabilities
- Made a dev tool to draw game assets in a click and drop fashion
- Made several dev tools to manipulate image data or import/export image arrays
- [In Progress] Working on generating surface normals on sprites for dynamic lighting