Brandon Do

do.brandon97@gmail.com | (423) 829-0495 | linkedin.com/in/brandon–do | github.com/brvndon-do

Skills

Languages: C#, C, C++, JavaScript/TypeScript, SQL

Frameworks: React, Angular, Node.js, .NET, Entity Framework

Technologies/Tools: Linux, Git, Azure, Github Actions, Docker

Education

Georgia Institute of Technology, M.S in Computer Science — Aug. 2024 – Present

University of Tennessee at Chattanooga, B.S in Computer Science — Aug. 2016 – Aug. 2020

Experience

Software Engineer, Steam Logistics – Atlanta, GA — Jan. 2023 – Oct. 2025

- Developed and maintained an internal TMS and reporting tool for brokers that integrated with Revenova/Salesforce
- Established a component library with documentation and style guides using Storybook, achieving UI/UX consistency across applications
- Integrated with 3rd-party services (Project44, CargoWise, and TruckerTools) to facilitate communication between brokers and carriers

Software Engineer, Carbon Five – Chattanooga, TN — May 2022 – Dec. 2022

- Worked and consulted with clients to provide engineering advice and technical solutions
- Developed and managed an internal application for allocations and timesheet tracking

Software Engineer, Future Capital – Chattanooga, TN — Mar. 2021 – May 2022

- Implemented a two-factor authentication system for user login, improving overall security for our clientele
- Lead full-stack developer for an advisor dashboard, which allowed financial advisors to efficiently manage wealth clients
- Integrated with financial providers into our platform for importing and exporting data regarding trades, holdings, etc.

Projects

b8 – Chip8 Interpreter
https://github.com/brvndon-do/b8

- Implemented a complete CHIP-8 virtual machine in C, including instruction fetch–decode–execute loop, memory model, registers, timers, stack, and keypad handling
- Decoded and executed CHIP-8 opcodes at the bit level using masking and shifting, adhering closely to the original specification and edge-case behavior
- Built a platform-agnostic rendering and input layer using SDL3 to emulate the original display and keypad interaction

black-orion – Web Game Engine
https://github.com/brvndon-do/black-orion

- Developed a browser-based game engine using TypeScript and three.js, focusing on real-time rendering and scene management
- Implemented the Entity-Component-System (ECS) architectural pattern to decouple game logic from data and improve scalability and maintainability
- Explored engine-level concerns such as data-oriented design, update loops, and separation of concerns