Matt Zhang

github.com/mzhang   matt@zhang.software   zhang.software   in mattzhang-

EDUCATION

University of Waterloo 2020 – 2024 | GPA: 91.4%

Bachelor of Software Engineering (BSE)

PROFESSIONAL EXPERIENCE

Spatial Systems Sep 2022 – present | San Francisco, CA

Software Engineer

- Developed cross-platform features for 3D environments in web/VR in **Typescript/React, Go** and **C#**
- Engineered cubemap-based controllable visualization of 3D spaces, reducing required bandwidth by **89%**
- Architected and implemented user avatar interactions system accumulating over **2 million** weekly uses
- Implemented chat services and custom keybinds leading to **23%** increase in average user retention

Immigrate.ai Dec 2021 – Apr 2022 | Toronto, ON

Software Engineer

- Created **React Native** cross-platform NLP chatbot portal resolving **87%** of user inquiries
- Authored full-featured task management platform in **React** to empower & organize 12 internal teams
- Optimized rendering & memoization, reducing API calls by **28%** and load time by **32%**
- Implemented travelling salesman heuristic to reduce average runtime by **27%**

Imagine Communications May 2021 – Sep 2021 | Toronto, ON

Fullstack Developer

- Built management dashboards for top international TV broadcasters using **React** and **Vue**
- Triaged & fixed critical authentication bugs in **C#** that blocked **35%** of users from accessing products
- Reduced complexity and file size of product deployment by **63%** using **WebSockets** and **RabbitMQ**
- Launched pipeline to normalize **PostgreSQL** databases, reducing storage by **25%** and query time by **33%**

PROJECTS

Automated Proof Checker ↗

- Designed & implemented scanner, validator and checker for mathematical proofs in **C#**
- Modified Shunting-Yard algorithm to parse n-ary inputs, reducing time complexity from cubic to linear

MIPS Compiler

- Built high-level programming language compiler converting Scala-like syntax to MIPS assembly
- Supports type checking, functions(nesting, scoping and recursion), closures and garbage collection

Modular Board Game Engine & CPU

- Architected & built turn-based game engine in **C++** supporting creation of arbitrary games at runtime
- Implemented minimax with alpha-beta tree pruning for AI capable of playing arbitrary user-created games

LANGUAGES & TECHNOLOGIES

Typescript   Python   Go   C++   C#   React/React Native   SQL   MongoDB