Hai Yang Tang

haiyang.tang@alum.utoronto.ca | Hai Yang Tang | Portfolio

EDUCATION

University of Toronto September 2019 - November 2023
Honours Bachelor of Science in Computer Science and CCIT

WORK EXPERIENCE

Intern Full Stack Developer (TA Application Portal) January 2022 - August 2022
University of Toronto

- Created **React** components and RESTful **Express** APIs to automate the assignment of TAs to course sections based on their availability, reducing the amount of time instructors spend scheduling TAs by **2-3** hours.
- Developed library code in **JavaScript** to fetch course data from a second party RESTful API and store it in a **PostgreSQL** database, removing the need for instructors to manually create **50+** sections per course when hiring TAs.
- Implemented responsive layouts in **15+** unique web pages using **Bootstrap 5**, enabling the portal to be used on many previously incompatible viewports including portrait browser windows and mobile devices.
- Added integration tests to the codebase's **Mocha** test suite, increasing code coverage to **~80%** and protecting the application from authentication and rate limiting related regressions.

MISC EXPERIENCE

Volunteer QA Tester March 2024 - Present
Hoplite.gg

- Created a Google Apps Script using **JavaScript** to automatically generate locale JSON files from design spreadsheets, reducing patch preparation time by **~1** hour.
- Found, documented, and retested **40+** issues across multiple updates, minimizing gameplay disruptions on the server.

PROJECTS

Hoplite Stats - Hoplite.gg Player Stats Viewer https://github.com/miosenpai/hoplite-stats

- Automated Minecraft client actions through the Mineflayer library, enabling the **Nuxt** backend to scrape player statistics from the in-game GUI.
- Implemented caching for statistics using **Redis**, reducing traffic generated from redundant scrapes by **60%**.
- Developed **SSE** endpoints which updates subscribers on the progress of ongoing scrapes, allowing the frontend to fetch stats right as it becomes available without introducing polling overheads.

Maleficus - Tower Defense Game https://ryn.moe/projects/maleficus

- Architected tower scripts for the game with another developer using **OOP** principles, allowing **15** different towers to be implemented while adhering to **DRY** best practices.
- Used **Unity**'s serialization system to expose appropriate parameters when developing various game scripts, enabling other team members to balance mechanics and swap assets without needing to modify the source code.

SKILLS

Languages	JavaScript, TypeScript, C#, Java, Python, SQL, HTML, CSS
Frameworks	React, Next.js, Vue, Nuxt, Express, ASP.NET Core, Unity, WPF
Tools	Git, Docker, GitHub, GitLab, Jira