+1 647 974 5597
vincent@zvikaramba.com
zvikaram
vince2678

Vincent Zvikaramba

Education

2021 **Honours Bachelor of Science**, *University of Toronto*
Statistics, Computer Science

Skills

Programming	Java, Python, (POSIX) Shell, Typescript/Javascript, C, SQL
Tools	Docker, Git, Node.js, Gerrit Code Review, GNUMake
Frameworks and libraries	Java Collections, Guice, Gson, Guava, Lombok, JUnit, Spring, React

Interests

Electronics I enjoy tinkering with electronics at any level. Particularly enjoyed working with FPGAs and Verilog in university for programmatic circuit design.

Reading I come from a family of avid readers; I'm usually reading research articles, documentation or spec sheets for miscellaneous electronic components.

Music I love listening to music and currently learning guitar (albeit very slowly).

Experience

2016–2017 **Teaching Assistant**, *Software Tools and Systems Programming*, University of Toronto

- Instructed students in tutorials and helped them troubleshoot and identify bugs in programming assignments
- Graded programming assignments and provided feedback for improvement
- Worked as part of a team of TAs to proctor tests and mark test papers

Projects

2022 **Client Management System**, *South-Asian Women's Rights Organization*, Toronto

- Worked as part of a team to build a client management system for SAWRO's client database
- Used Python and Flask to write the web application
- Used an ORM (SQLAlchemy) to interact with the SQL database

2022 **Website**, *Promatec Solutions*, South Africa

- Collaborated with another developer to build a website and e-mail solution for Promatec
- Used docker for containerisation and easy deployment of services
- Deployed and leveraged Gerrit Code Review for collaboration and code review
- Wrote a python script hooking into the CloudFlare API for managing DNS records required or used in application containers
- Used Node.js as web server and as the backend for APIs exposed to the frontend
- Used React.js for UI component design and reuse

2019–Present **Automated MMO Client**, Toronto

- Decompiled and debugged game client code to identify obfuscated game code functionality for modification via injection
- Used reflection to inspect game state and instantiate some objects at runtime
- Used injection to expose game client classes, fields and methods where possible; and to add to or override original game client code
- Used Swing and AWT to design UI elements and intercept input events in order to inject new or modified input events to the game client
- Used breadth-first search and A* search with custom heuristics to create path-finding algorithm for searching optimal game world paths

2015–2019 **Android Custom ROMs**

- Maintained support for Samsung devices originally running Android versions 4, 5, 6.0
- Maintained legacy kernel code and backported new kernel code
- Developed tools and user-space interposer libraries for forward compatibility of proprietary libraries and programs with future android releases
- Maintained Makefiles for use in the build system
- Deployed Gerrit for code review and collaboration and Jenkins for builds and continuous integration