VINCENT ZVIKARAMBA

Toronto, ON, Canada| 647-974-5597 | zvikaram@mail.utoronto.ca
GitHub: https://github.com/vince2678 | LinkedIn: https://linkedin.com/in/zvikaram

EDUCATION

Honours Bachelor of Science Statistics and Computer Science 2014 - 2021
University of Toronto

- Courses: Software Tools & Systems Programming, Data Analysis, Software Design, Computer Organisation, Algorithm Design and Analysis, Software Engineering, Artificial Intelligence, Operating Systems, Graph Theory

SKILLS & ABILITIES

- Programming: Java, Python, C/C++, JavaScript, SQL, HTML, Unix shell scripting
- Tools: Git, Docker, Gerrit Code Review, Jenkins, Build Kite, Gradle
- Frameworks and Libraries: Java Collections, Guice, Gson, Guava, Lombok, Junit, Spring, Flask
- Versed in Object Oriented Programming techniques, Agile methodologies, multi-threaded programming, unit testing, debugging

EXPERIENCE

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

- Instructed students during tutorial sessions on course material and aided them troubleshoot and identify bugs in programming assignments
- Graded programming assignments and term tests and provided feedback for improvement

RELEVANT PROJECTS

Team Leader, Computer Organisation, *University of Toronto* 2016

- Designed a basic video game on an Altera DE-1 SoC
- Used Verilog HDL to design the circuitry of the game, which consisted of different modules in VHDL for a pseudo random number generator, VGA controller, RAM and rate divider; and ModelSIM to run simulations and verify the design

Team Member, *Introduction to Software Engineering* Project, *University of Toronto* 2018

- Collaborated with six team members to write and design a location-based chat room application in Java for Android
- Used Agile methods for software development and completed three sprints in one month
- Implemented Gerrit setup with GitHub as a backup for version control and code review

Individual, *Web-based media player* 2019

- Used PHP to implement server-side code for parsing media metadata and interfacing with the server-side database
- Used PHP to create API which issues JSON-formatted data for use in client-side user interface
- Used SQL to design database schemas and relations; and MySQL for database storage
- Used CSS and HTML for player design and JavaScript for active client-side user interface

Individual, Automated *MMO Client* 2019-Present

- 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