Michael Kwan

647-882-8209 ◇ michaelkwan1238@gmail.com ◇ https://github.com/Phys09 ◇ https://www.linkedin.com/in/michael-k-kwan/

Industry Interests: Software Development, Software Engineering, Computer Programming, IT Services, Technology, Games Programming and Modding

Skills

Languages: Python | C++ | C | C# | Java | Javascript/Typescript | HTML | LaTeX | Markdown | R

Libraries/Frameworks: Numpy | Pandas | Matplotlib | Scikit Learn | Flask | Sqlite3 | React | JavaFX

Tools/Technologies: Git | Github | Docker | Visual Studio | Vim | Eclipse | Google Collab | Jupyter | RStudio | Pycharm

Operating Systems: Windows | Linux/Unix

Familiarity With: Neo4j | PostgreSQL

Education

University of Toronto Mississauga, Honours Bachelor of Science - Mathematics 2024

- Mathematics Major, minors in Computer Science and Statistics
- Relevant courses in Software Engineering, Software Design, Machine Learning, Data Structures and Algorithms, Systems Programming, Theory of Computation, Computer Organization, Data Analysis, Statistical Learning

Relevant Experience

Captain Beacon Cooldowns and Tweaks (C#, Unity Game Engine) **Summer 2024**
Software Developer

- Project Link: http://thunderstore.io
- Extended functionality of existing software using community-made technical documentation, software development kits, provided decompiled source code, and other modding tools, resulting in over 6325 downloads since release
- Enabled users to customize existing character effects and cooldowns through a user-friendly plaintext configuration option, enhancing the overall player experience
- Conducted root cause analysis of the application through decompiling, testing, experimenting, and debugging of existing game code, and modified game code, culminating to a deeper understanding of system logic and the production of maintainable code
- Provided ongoing support post-release by maintaining and updating the project as needed
- Engaged with users on GitHub, promptly addressing issues, inquiries, and bug reports while offering clear updates on the project's status

SigmaTech Banking Application (Javascript, CSS, React, PostgreSQL) **Winter 2022**
Frontend Developer

- Collaborated in a team of 7 to create the frontend and backend of a banking web application following the SCRUM software development life cycle
- Designed and built a React frontend with a PostgreSQL and Node.js backend, enabling users to manage account information, transfer funds, and apply for loans
- Created and designed interactive web pages that effectively communicated with backend database endpoints

Data Visualization Tool (Python, Flask, React, CSS) **Fall 2022**
Fullstack Developer

- Worked in a 3-person team to develop a data visualization web app using React, displaying datasets through graphical representations
- Developed a backend server with Python and Flask, implementing SQLite (using SQLite3) to store data for access via API endpoints
- Contributed by creating backend endpoints, facilitating data retrieval for the frontend display

Careconnect (Javascript, React, CSS) **Spring 2023**
Fullstack Developer

- Developed a ReactJS web application during the 2023 Deerhacks hackathon, pairing users with personal support workers (PSWs) based on various criteria
- Responsibilities included designing and formatting web pages, as well as debugging and testing algorithms

Othello Board Game (Java, JavaFX) **Fall 2019**
Software Engineer

- Implemented the game Othello in a team of 5, adhering to the SCRUM software development cycle, with the Java programming language
- Developed a full GUI using the JavaFX library, featuring animations and options to play against other players or AI opponents

Other Experience

Advent of Code 2024 (C++, CMake, Ninja Build Tools) **Winter 2024**

- Solved a variety of algorithmic challenges using modern C++ (C++17/20), emphasizing performance and readability
- Utilized knowledge of data structures like hash maps, arrays, and trees along with algorithm techniques like recursion and multiple pointers in order to implement optimal solutions for problems while documenting written code and setting up tests
- Adhered to software engineering best practices leveraging debugging tools to understand and write clean, modular, well-documented code, and test driven development to ensure maintainability and efficiency

Arduino Alarm System (C++)
C++ Developer

- Implemented and programmed a custom alarm system using an arduino microcontroller in C++, interfacing with physical components such as LCDs, RFID Readers, LEDs and Buttons on a circuit board
- Used Electronic Design Automation and Planning software 'fritzing' to draft a proof-of-concept design for a custom alarm system, then built it
- Produced a design doc outlining required parts and materials needed, diagrams of the circuitry, and pseudocode needed for building

6 Degrees of Kevin Bacon (Java, Neo4j, Dagger2) **Winter 2022**
Java Developer

- Developed a program connecting actors through Kevin Bacon within six degrees of separation, utilizing Java and the Neo4J graph database
- Leveraged Google's Dagger2 dependency injection framework, integrating Neo4j for data access and managing server request handlers by supplying program components when needed

Simulated File System (C) **Winter 2020**
Systems-Level Developer

- Created a simulated file system using C and binary I/O functions, allowing users to create, delete, and search for binary files
- Implemented file operations that mimicked real file system behavior, reading and writing to files, enhancing understanding of data storage and access in Linux environments

Paint Application (Java, JavaFX) **Fall 2019**
Java Software Engineer

- Designed a paint application in Java using object-oriented design patterns such as Model-view-controller (MVC), and the Factory patterns, letting users create digital artwork on a desktop application
- Developed a user-friendly GUI with the JavaFX library, providing various brushes and tools for painting

Pythonic RPG (Python) **Summer 2019**
Game Programmer

- Developed a top-down RPG game using Python 3 and the Pygame library, which had players navigate levels and complete objectives
- Created interactive game elements such as walls, enemies, and collectible items, enhancing gameplay experience through the clever use of having the player make the right decision at the right time