THOMAS VY

vythomas97@gmail.com • (403) 389-4180 • thomasvy.github.io/Website
www.linkedin.com/in/thomas-vy • www.github.com/ThomasVy

A motivated software engineering graduate with an interest in web development and a background in automation and build systems. Currently seeking a permanent full-time software developer position with the possibility to evolve into a leadership role.

SOFTWARE SKILLS

PROFICIENT SOFTWARE: C/C++, Python 3, JavaScript, Java, Git, React/TypeScript, MySQL, RESTful APIs, HTML/CSS.

FAMILIAR SOFTWARE: Node.js, PowerShell, Batch, Bash, PHP, Laravel, OpenGL, Squish GUI Tester, Qt, XPath, Team Foundation Version Control.

EDUCATION

MAY 2021

BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING, UNIVERSITY OF CALGARY

- Graduated with Distinctions and Internship Program, 3.89 GPA.
- Completed courses in Data Structures & Algorithms, Operating Systems, Data Base Management Systems, Graphics Programming, Networks, and Computer Security.

EXPERIENCE

MAY 2019 – AUGUST 2020

SOFTWARE DEVELOPER INTERN, GEOSLOPE INTERNATIONAL LTD.

- Designed and developed UI dialogue boxes and control code using MVC, C++, and MFC Library.
- Increased UI test coverage by developing Squish GUI tests and integrating them into the nightly builds which led to an increase of UI bugs being caught.
- Modernized build systems by moving builds systems from Team Foundation Server to Azure Pipelines which led to faster build times.
- Taught colleagues Git/GitHub by presenting a lunch and learn which led to a
- Increased team's efficiency by implementing PowerShell/Batch scripts to automate check-in testing which led to a decrease in bugs being merged into the main branch.
- Obtained a solid understanding of leadership skills by acting as a stand-in team lead for stand-ups, backlog grooming, retrospective, and sprint planning.

OCTOBER 2018 – JANUARY 2021

SOFTWARE TEAM MEMBER, UNIVERSITY OF CALGARY'S SOLAR CAR TEAM

- Upgraded car's functionality by helping develop a song player in Qt and C++ which led to a deeper understanding of planning.
- Successfully wrote documentation for porting software code from Linux to Raspberry Pi by testing various methods and modifying code that enabled code to run in the car.
- Educated recruits on Git/GitHub by performing a Git/GitHub presentation which led to recruits being able to properly perform version control tasks.
- Successfully expanded the software team by developing interview questions and performing recruitment interviews.

MAY 2018 – AUGUST 2018

SUMMER RESEARCHER ASSISTANT, ROBOTICS AND SENSOR NETWORK GROUP

- Obtained an understanding of automation by creating a navigation system written in C++ and Python which led to a robot traversing a room autonomously.
- Developed a mapping program for the robot by converting LIDAR data to a visual representation of the room using C++.
- Improved my communication and presentation skills by teaching colleagues how to use ROS, C++, and Python.

SOFTWARE PROJECTS

MOSHIRLEARNING, github.com/ThomasVy/MoshirLearning

- Designed and developed a server-client application that allows teachers and students to access their courses (Similar to the D2L website). Teachers can edit their courses, add new courses, and manage students in courses.
- Written using Java, MySQL, and network sockets.
- Received an outstanding final project reward in 2018.

SPACESHIP GAME, github.com/ThomasVy/Spaceship-game

- Created a game that controls a spaceship to collect gems scattered randomly on the screen. The game ends when all gems are collected without touching a fire object.
- Uses mouse clicks and WS keys to navigate the spaceship around the map.
- Uses C++, matrix manipulation, and OpenGL.

P2P MESSAGING APP, github.com/ThomasVy/P2P-Messaging-App

- Designed and developed a messaging application that uses peer-to-peer networking to communicate with others.
- The application initially connects to a registry to collect the peer information list. After collecting the peer information, the program then periodically sends peer-to-peer alive messages to notify other users of the existence of the program. Then users' broadcasts messages to all the live users.
- Written using Python 3, TCP/UDP sockets, and concurrent programming.