

CLINT GALVEZ

Mississauga, ON

(647) 878-9839 - CLINTGALVEZ@gmail.com - <https://clintgalvez.com/>

EDUCATION

Honours Bachelor of Computer Science,
Computer Game Development, Co-op Option
Carleton University, Ottawa, ON

September 2020-April 2025

- Fourth-year standing, 11.07/12 (3.9 in 4.0 GPA)
- Honours: Dean's Honours List

TECHNICAL SKILLS

Languages C/C++, C#, GDScript, Python, Java, JavaScript, HTML, CSS, VB

Tools Git/GitHub, Unity, Godot, Visual Studios, IntelliJ, QT Creator, Oracle VM VirtualBox, Linux/Ubuntu, Jenkins, SVN, Bitbucket

Concepts OOP, FSM, Space-Time Complexity, Recursion, Simulation, Algorithms & Data Structures, Multi-threading, Pipelines, Agile

WORK EXPERIENCE

Software Automation Engineer
Lumentum Operations LLC

May 2023-December 2023

- Spearheaded the integration of continuous integration/continuous deployment (CI/CD) pipelines, streamlining the software development lifecycle
- Collaborated with cross-functional teams to identify and resolve software issues, contributing to the timely delivery of high-quality products
- Implemented efficient performance testing procedures, optimizing software responsiveness and enhancing system reliability
- Conducted code reviews and provided constructive feedback to team members, fostering a culture of continuous improvement
- Utilized industry best practices to create and maintain documentation for automated test procedures, ensuring knowledge transfer within the team
- Actively participated in Agile development methodologies, contributing to sprint planning, retrospectives, and maintaining a rapid development pace

PROJECTS

Dark Souls Clone (Unity) [C#]

Available on GitHub

- Worked with character models, animations, materials, camera handling and collisions
- Implemented weapon swapping and attack combos for the player

Cranial Electrotherapy Stimulation (CES) Device Simulation (Qt Creator) [C++]

Available on GitHub

- Developed and maintained a complex software project using object-oriented design principles such as abstraction, inheritance, and polymorphism
- Utilized design patterns such as factory pattern, observer pattern, and decorator pattern to improve code structure, encapsulation, and flexibility

Elevator Control System (ECS) Simulation (Qt Creator) [C++]

Available on GitHub

- Designed and developed a multi-threaded elevator control system (ECS) application using C++ and QT framework; along with functional simulated elevators and an interactable user interface to test multiple elevator request-handling strategies
- Implemented various state handling mechanisms such as emergencies, door obstacles and elevator overloads
- Utilized signals and slot mechanisms for communication between elevators, floors, and the control system for a smooth operation
- Created use cases, diagrams (ie. UML, sequence, state), traceability matrix, and a manual for project documentation

Multi-Threaded Probability Simulation [C]

Available on GitHub

- Incorporated multi-threading in C to simulate multiple randomized executions/runs of a battle between two heroes and a pirate crew and retrieve statistics on the given results

Multi-Process Library Server [C]

Available on GitHub

- Implemented a socket-based server to handle client requests for a library management system, in which users can check books in/out as well as view and add books to the library

- Utilized signal handling to gracefully close the server and release resources in the event of an interruption
- Employed send and receive data functions to transfer information between client and server via socket connection efficiently

Ant Wars Simulation [Java]

Available on GitHub

- Implemented a simulation using Java GUI that generates a visual display for users
- Designed the program using OOP principles (ie. abstraction, encapsulation, inheritance, polymorphism) to prove knowledge and understanding of said principles
- Applied abstraction and inheritance in the ant class to hide common-code shared by its subsidiary classes (ie. queen and worker), making each class unique and specialized, resulting in easier debugging as problems can be deduced to a specific class
- Incorporated polymorphism in the simulation class to simplify tracking of each instance of the queen and worker classes through the use of their abstract parent class (ant), resulting in simplified and easier-to-understand code
- Compiled an analysis report on the project detailing the space/time complexity of the implementation, a quality analysis of any trade-offs, and areas of improvement

PokeBattler [Python]

Available on GitHub

- Utilized written communication skills to explain the project proposal to the instructor
- Implemented web-scraping to keep the game up-to-date with the newest Pokemon upon each launch

Sheridan Game Competition

October 2018-November 2018

- Implemented the base code
- Collaborated well with teammates to debug and find solutions to problems
- Sourced through the final code to find areas to further enhance/optimize the game