Mitchell Ma

604.445.2665 | mitchellma0427@hotmail.com | mitchma.bitbucket.io

EDUCATION

UNIVERSITY OF BRITISH COLUMBIA | BSc in Mathematics and Computer Science
September 2014 - May 2018 | Vancouver, BC

VOLUNTEER & WORK

TUTOR
September 2013 – Present

- Specialize in teaching mathematics and computer science (up to third year university level)

PROJECTS

MIND THE GAP | Android application for London's transit system
July 2016 – August 2016 | Solo | Java

- Implemented parser for London's transit API that transforms JSON and XML holding the transit information
- Integrated Google Maps into the Android application
- Coded up UI that marks out transit lines and station locations and provides additional information when selected
- Created UI pages that show train schedules for the station selected by the user
- Tested major functions using JUnit with edge and practical cases

INSIGHTUBC | Course and building explorer for UBC
January 2017 – April 2017 | 2 members | HTML • SCSS • TypeScript

- Created functions to extract and filter out contents from UBC Pair's ZIP data
- Implemented ability to cache extracted ZIP data onto local disk as JSON files after removing unnecessary information
- Wrote up front end web page that allows users to query UBC's buildings and courses as well as to create course schedules given a subset of courses and buildings
- Set up Node server with Restify as middleware for routing HTTP requests
- Coded rudimentary database in the server with basic DDL and DML that support custom SQL-like queries with basic match, find, and aggregation
- Created functions that transform user searches into queries and displayed search results in friendly manner on front end webpage
- Extensively tested the back end using Mocha (mix of edge and practical cases)

GSMS | Grocery store management system
September 2017 – December 2017 | 4 members | HTML • SCSS • TypeScript • (PL/)SQL

- Source code and demo instructions are on GitHub and linked via my website
- Using ERDs, designed BCNF database suitable for grocery-related queries and operations (applicable for other scenarios)
- Coded front end webpage with HTML, SCSS, and JQuery to act as graphical command panel for customers and employees
- Wrote RESTful server back end powered by Node that directly receives user input from the front end and sanitizes the data
- Set up Node server with Restify as middleware for routing HTTP requests and hooked up server with remotely hosted Oracle database
- Implemented and tested the conversion from user input data to SQL statements suitable for Oracle database and tried out PL/SQL for performance improvements
- Displayed the database's response in a clear and user-friendly manner in the front end webpage and allowed for further filtering and searching of the returned results

POST ME | Mobile app to post anything locally
December 2017 – March 2018 | 2 members | React Native • Java • Swift • SQL

- Responsible for Android-side testing and verification
- Integrated navigation between the sign-in page, main application screen, and settings
- Coded up custom, reusable components such as toggles necessary for settings page
- Wrote up the messaging UI and system that can send and receive push notifications from server
- Implemented user authentication through email, Google, and Facebook using Google's Firebase

- Set up RESTful server using Express as middleware for routing HTTP requests and connected server to PostgreSQL database
- Server verifies and signs in users, transforms data from user HTTP requests, and stores and retrieves data such as posts and messages from PostgreSQL
- Tested some aspects of the code using Jest

SCRAPEYARD | Crawler and scraper for The New York Times

January 2018 - January 2018 | Solo | Python 3

- Source code is available on Bitbucket and linked via my website
- Coded single-threaded web crawler that runs asynchronously to maximize URL requests
- Included option to limit number of pages crawled as well as speed at which webpages are accessed to allow for long-term, background operation
- Implemented multiprocess scraper for loading the websites provided by the crawler and extracts specific information by targeting specific tags in the HTML
- Extracted data are transformed into human-friendly JSON files using multiprocess producer-consumer pattern and are stored on disk (also suitable for NoSQL database)
- Analyzed performance gains (and surprising performance degradation depending on hardware) of single-threading, asynchronous pattern, and multiprocessing

SPIT! (GLOBAL GAME JAM 2018) | Local multiplayer shooter game

January 2018 - January 2018 | 5 members | C#

- Source code is available on Bitbucket and linked via my website
- Designed and brainstormed game theme and mechanics, resulting in a third-person shooter in which the player helps a mutated monster transmit disease using its infectious, long-range spit (GGJ2018 theme was transmission)
- Prototyped game using Unity and tested various features such as movement control and map generation
- Coded map and level creation by procedurally generating maze blocks and obstacles
- Responsible for randomly spawning power-ups with various effects and interactions with players
- Implemented and tested multiplayer using UNet for power-up, map obstacle, and player location synchronization

ONCE UPON A TIME | RPG with focus in farming, crafting, and adventuring

January 2018 - Present | Solo | GDScript • Python 3 • C++

- Try out the debug copy on my website
- Responsible for writing up multiplayer code for synchronizing player, item, and map obstacle locations using Godot's ENet
- Set up an extremely customizable equipment system in which the player can change the colour of any part of the equipment through RGB sliders
- Implemented interactions such as chopping, hammering, and scything various map obstacles as well as basic farming mechanics such as tilling soil and planting seeds
- Finished save system that records progress and connected peers by reading and writing JSON to disk
- Created, drew, and rigged all assets using Blender and Illustrator and further polished in Godot using shaders and materials

LOST | Grid-based pathfinding program

May 2018 - June 2018 | Solo | C++

- Source code is available on Bitbucket
- Integrated SFML and TGUI and set up the graphical, windowed application
- Coded user input handler to easily allow for grid alteration such as adding obstacles and moving start and terminal points
- Implemented A* search and tested multiple data structures to effectively cache the open and closed lists
- Extended A* class into Jump Point Search that expands less cells and provides the optimal path in shorter period of time
- Included options to use different heuristics such as octile, to allow for pathfinding animation, and to disable corner cutting for more natural paths

TECHNICAL SKILLS

Familiar with:

Java • TypeScript (JavaScript) • Python 3 • HTML & SCSS (CSS) • LaTeX • C • C++ • Rust • MATLAB • (PL/)SQL • NoSQL • PHP • Assembly • Visual Basic

I've used these before:

JUnit • Mocha • Jest • Node.js • React Native • Unity • Godot • Adobe Illustrator • Adobe Photoshop • JetBrains IDE • Sublime Text