UBC Science Co-op

T: 604.822.9677 | F: 604.822.9676 | science.coop@ubc.ca | www.sciencecoop.ubc.ca

James Graham

***Phone**: (204) 298-6348 | **Email**: jgraham.paul@gmail.com*
***GitHub**: https://github.com/jamesgraham0 | **Portfolio**: https://jamesreactportfolio.netlify.app*

EDUCATION

University of British Columbia (UBC), Vancouver *Expected 05/2024*

- BSc. Computer Science
- Member of the Varsity Men's Soccer team from 2018-2021 *Academic All-Canadian Award 2020/2021*

PROJECTS

Note Sharing Application *June - Current*

MongoDB | Express | React | Node.JS | Typescript | Firebase

- A full-stack, social-media web application intended for uploading and sharing notes with other students.
- User authentication, storage of image files, and metadata for notes implemented using Firebase and MongoDB.
- React and Redux used to manage state.
- Implemented user profiles, image uploading, editing, and deleting, a search feature, and a personal feed based on a note's rating out of five stars, relevance to a user, and filters.

SIFT Keypoint Matching, Neural Networks and Deep Learning *March - 2022*

Python | PyTorch | NumPy | OpenCV | MatPlotLib

- Implemented SIFT keypoint matching with and without RANSAC in order to match scale and rotationally invariant features between two images.
- Implemented various fundamental deep learning layers of PyTorch using NumPy arrays. Utilized deep learning tools such as activation functions, linear layers, max pooling layers and convolution layers.
- Experimented with different hyperparameters on an image classification task to find the best hyper-parameters.
- Experimented with PyTorch model zoo and Mask R-CNN- an image segmentation task which assigns a label to every pixel rather than labeling the class of a bounding box.

Tower-Defense Game *Aug - 2021*

Python | JSON | Gimp

- Used Pygame API and OOP techniques.
- Found a unique solution to animating sprite sheets by separating them into smaller images through JSON and iterating through each image.
- Implemented waves enemies that follow the path by simple use of arrays and iteration. Each enemy triggers the action of towers and collision detection with bombs based on distance from towers.
- Implemented dynamic colour highlighting when placing towers.
- All sprites except for the turrets were drawn by myself using gimp photo editing software.

WORK EXPERIENCE

UBC, Vancouver — *Fitness Centre Floor Staff* *Summer 2022*
Under the GUI Academy Instructor, Vancouver — *Python / Robotics / Gimp* *Summer 2021/2022*
UBC, Vancouver — *Camp Counselor* *Summer 2019*
Colosseo, Winnipeg — *Server* *Summer 2018*

TECHNICAL SKILLS

Programming Languages: Python, Java, C, C++, Typescript
Web: HTML, CSS, Javascript
Other: Git, MongoDB, REST, SQL
Testing: JUnit, Chai
Frameworks: React, Node.JS, Express