Hao Li

Phone: +1 (438) 927-8699 | Email: hao.li7@mail.mcgill.ca | GitHub: github.com/lhcaleo

Education

Bachelor of Science in Computer Science **September 2016 – January 2021**
McGill University, Montreal, Canada

Skills

Languages: Fluent in English, Basic French, Fluent in Chinese
Programming Languages: Java, Python, C++, C, C#, HTML, CSS, JavaScript, Bash, MATLAB, Assembly language (MIPS)
Tools: Linux, Git, Unit Test, Make file, PApplet, Scikit-learn, TensorFlow, Natural Language Toolkit
Core Courses: Algorithms and Data Structures, Computer System, Software System and Design, Artificial Intelligence, Concurrent Programming, Applied Machine Learning, Natural Language Processing, Probability, Discrete Structure, Numerical Method

Software Projects

Binary Classification with Linear Models in Machine Learning **August 2019 – September 2019**

- Implemented Logistic Regression using gradient descent and Linear Discriminant Analysis models from scratch **in Python**
- Performed binary class classification on two datasets while each data set contains more than 10 discrete features
- Applied k-fold **cross-validation** with randomly shuffled training and testing sets to unbalance the distribution of classes
- The accuracy of Linear Discriminant Analysis model is **94%** which is about 10% higher than Logistic Regression Model

Interactive Earthquake Map of the World **July 2019 – August 2019**

- Built an Applet of a 2D earthquake map of the world using Processing (**Java**) graphics methods and Unfolding Maps library functions to visualize earthquake data
- Collected recent earthquake and city data from live RSS feeds and plotted corresponding land quake, oceanic quake, and city markers on the map depending on parsed properties with a key besides the map
- Constructed markers with different shapes and colors according to their magnitude, category, depth and time
- Created abstract classes and interfaces to organize all marker classes for class hierarchy and polymorphism
- Built event handlers to interactively respond to mouse input to hide and show city markers affected by an earthquake or all the earthquakes that can threat the clicked city, to keyboard input to switch among three map providers

Enhanced Copy Button and Selection Search **June 2019 – Present**

- Built a Chrome extension to automatically create copy buttons for tags that represent code blocks in GitHub and Stack Overflow
- Combined Bootstrap tooltips with **CSS** code to display different button styles with the help of mouse event listeners and **jQuery**
- Enabled users to search selected text in multiple search engines such as Wikipedia and Bing in the form of new browser window to enhance fast search and user experience
- Created context menu items for user to choose search engines or read out the selected text with the support of text-to-speech API
- Designed a popup HTML page that shows a quick user guide of the extension when the icon on the toolbar is clicked

Pentago-Swap AI Player **March 2019 – April 2019**

- Developed an AI agent for Pentago-Swap which is a 2-player board game that consists of two phases: placing and swapping
- Implemented **Monte-Carlo Tree Search algorithm** to do random simulations and reduce the number of moves to be evaluated
- Combined Minimax algorithm and evaluation function in the expansion step of MCTS to optimally select the best move with the help of Alpha-Beta pruning approach to increase the efficiency of game tree searching
- Created additional strategies as AI chooses to place the first piece to any one of the four available center position
- Search and test one step ahead before the calculated “best move” is placed to ensure that there are no other better moves as well as preventing the opponent to win right after my best move
- Competed with random and 290 student players and ranked in the top 25% in class tournament

Image Convolution **December 2018 – January 2019**

- Created various image effects such as edge and line detections by performing convolutions on input image
- Implemented different kernel computations to compute every output pixel value without boundary problem and data races
- Improved program performance efficiently by 53% by implementing multithreads (8 threads) in **Java**

Activity

McGill Wave Soccer Club **September 2016 – May 2020**

- Managed team training sessions on campus and participated in three seasons of Soccer Intramurals of McGill