DANIEL TREICHEL

355 Vista Glen Dr, Apt. 1303, Plano, TX 75074 • (512) 363-0747 • djt150030@utdallas.edu

LinkedIn: https://www.linkedin.com/in/daniel-treichel/

GitHub: https://github.com/Duallight

Skills

- C++
- Java
- Python
- Jupyter Notebooks
- JavaScript
- HTML/CSS
- Node.JS
- React
- Git
- MySQL
- Object Oriented Programming
- Agile

Education

Bachelors: Computer Science, Spring 2020
The University of Texas At Dallas - Richardson, TX

Projects

Vessel Identity Classifier using Machine Learning

- Worked on a team of five, with guidance from mentors at Raytheon Technologies, to improve a classifier that would predict a ships class from data taken from ships Automatic Identification System. We started with a classifier that provided 31% accuracy, and improved it to a 92% accuracy rating.
- The goal of this project was to prevent fishing vessels declaring their ship classification incorrectly allowing them to fish in illegal areas.
- Challenges included validating and cleaning large amounts of data, refactoring existing architecture to allow expansion, and team communication during the pandemic.

Curiosity: a game used to demonstrate server/client security

- Worked on a team of four to build a multiplayer game where one team would try to get a robot through a maze on a server, where everyone gets a chance to move the robot. The adversary team would have to hack the communications between the team and the robot and guide the robot to a game over area. The difficulties set were different public-key systems that determined how easy or difficult it would be to break.
- Some challenges we faced were making the difficulty to break encryption not too easy, and not impossible on various settings, making a front-end that was user friendly, and making a control system that allowed multiple people to control one character.

Text Editor

- Built a text editor using Java that had all the basic text editor features such as saving and loading a file as well as searching the file with or without regex.
- The most difficult part of this project was creating a responsive user interface that would not freeze when searching in a large file, which was solved by implementing multi-threading.

Simulated Computer System

- Built a simulated computer system consisting of a CPU and memory. This project was designed to give me a deeper understanding of the communications between the CPU and memory.
- The project was written in C++ and uses forks and pipes to communicate between the CPU and memory. The “CPU” in the project initializes a memory system and the pipes to communicate to it. It uses I/O f-streams to read instructions that are stored in the memory system, which is provided by sample text files. The output of the instructions is written to the screen.
- The main challenge of this project was getting the fetch and execution phases of the CPU to work efficiently, and handling interrupts in the CPU effectively.