

Yue Yang

919-937-4868 | yueyang0115@outlook.com | Durham, NC 27705

[LinkedIn: Yue Yang](#) | [GitHub: yueyang0115](#)

EDUCATION

Duke University

Master of Science, Electrical & Computer Engineering

GPA: 3.75/4.0

Expected in May 2021

University of Science and Technology Beijing

Bachelor of Engineering, Measuring & Control Technology and Instruments

GPA: 3.85/4.0

Sep 2015 - June 2019

SKILL

Programming Language: Java, C++, C, Python, Verilog, MATLAB

Tools: PostgreSQL, Docker, Gradle, JUnit, Hibernate, Mockito, JavaFX, Django, HTML/CSS, JavaScript, Bootstrap, CI/CD, Git, Emacs, Valgrind, GDB, Linux Environment

PROFESSIONAL EXPERIENCE

Software Intern (Software Engineering)

Duke University

May 2020 - Aug 2020

- Developed an Android game application which enables players to explore the virtual world, add friends, build buildings, buy equipment, fight monsters, upgrade and revive soldiers while walking around in real world.
- Stored data in **Postgres** database and used **Hibernate** ORM to interact with it. Resilient to failures/easy recovery.
- Used **TCP socket**, **JSON** and **Netty** for **Server-Client** communication and crafted a protocol specification for that.
- Wrote unit tests in **JUnit** framework and used **Mockito** framework to mock external dependencies in tests.

Research Intern (Medical Image Analysis, ML)

Beijing Institute of Technology

Aug 2018 - Mar 2019

- Worked on MRI data harmonization, developed a **CNN-based machine learning** algorithm and a linear mapping algorithm with Python. The proposed algorithms eliminate the changes caused by scanners while maintaining the fiber orientation in human brain. The harmonized data can be used for further medical analysis.

PROJECT

Mini Amazon System (Java, Python, Django, PostgreSQL, Protocol Buffer)

Apr 2020

- Developed a full-stack web application modeling Amazon system paired with warehouse and delivery system in Django. Simulated the whole process from buying products to getting the package delivered.
- Developed backend server with Java. Stored data in PostgreSQL and used Django ORM to interact with it.
- Used Google Protocol Buffer to communicate with world-simulator and mini UPS system partner.
- Designed UI with CSS, HTML and Bootstrap, realized full-featured web pages with search bar and shopping cart.

Risk Game (Java, TCP Socket, Multi-Thread, JSON, CI/CD, Docker)

Mar 2020 - Apr 2020

- Developed a game which enables users to attack territories and obtain resources, move and upgrade soldiers, chat and ally with other players. Developed backend server with Java and frontend UI with JavaFX and MVC.
- Added concurrency to deal with multiple players. Used TCP socket and JSON for Server-Client communication.
- Developed graph algorithm to implement path generation and resource allocation in game-world in better ways.
- Applied Agile methodology, carried out issue tracking and CI/CD pipeline, drew UML diagram and prototype.

HTTP Caching Proxy Server (C++, Network, TCP Socket, Multi-Thread)

Feb 2020

- Established a HTTP caching proxy server which can handle GET, POST, CONNECT requests.
- Used C++ 11 to implement RAII technique and modeled class with strong exception safety guarantee.
- Added concurrency to handle requests from different endpoints. Used TCP sockets to send and receive packets.
- Improved efficiency by caching responses according to validation and expiration rules in RFC7234.

Ride Sharing Service Web-App (Python, Django, PostgreSQL, Docker)

Feb 2020

- Developed a full-stack web application modeling Uber system with Django and PostgreSQL.
- The web-app enables users to login/logout, request/search/select/drive for/join in rides, change/view order status.
- Built user-friendly web-page with CSS, HTML and Bootstrap Library and used Docker for web-app deployment.

Thread-Safe Memory Allocator (C, Thread-Safe)

Feb 2020

- Implemented Malloc Library, experimented with First Fit and Best Fit allocation strategies and evaluated tradeoffs.
- Made it thread-safe, with a locked version (pthread mutex) and a lock-free version (Thread Local Storage).

Mini Command Shell (C++, Multi-Process, Valgrind)

Dec 2019

- The shell can launch programs, change working directory, provide access to variables, redirect and process pipeline.
- Implemented RAII technique and used Valgrind for memory leak detection.