

---

# JACOB LANE MOORE

---

Huntsville, AL 35806 ◇ jlmoors001@gmail.com ◇ (256) 701-7888

## EDUCATION

---

**MS, Georgia Institute of Technology**, Computer Science Expected 2027

- **GPA: 4.0/4.0**
- **Coursework:** Hypervisors, Distributed Systems, Software Architecture, Compilers

**BS, Mississippi State University**, Computer Engineering Graduated

- **GPA: 4.0/4.0**
- **Coursework:** Embedded Systems, Systems Programming, Computer Architecture, Operating Systems

## ABOUT ME

---

<b>Languages</b>	C++/C, Python
<b>Development</b>	Neovim, VSCode, Linux, Docker, CMake, Make, Zsh/Bash, Git
<b>ML/DL</b>	PyTorch, TensorFlow, RLLib, Scikit-Learn, Pandas, Grafana, Matplotlib
<b>Certs</b>	DeepLearning.AI, Neural Networks and Deep Learning (Andrew Ng), CompTIA Security+

## EXPERIENCE

---

**Software Design Engineer**, Torch Technologies – Huntsville, AL 05/2023 – Current

- Engineered advanced C++ microservices and libraries for a reinforcement learning (RL) ecosystem, integrating AI models into AFSIM-based defense simulations under logical or real-time clocking.
- Contributed to designing and implementing a plugin-style architecture (C++ and dynamic DLL loading) for RL observations, rewards, and actions, enabling on-demand extensibility and configuration-driven workflows.
- Developed ONNX model abstraction layers and data model libraries to streamline batch inference and complex data transformations, leveraging design patterns (factory, strategy, singleton) for maintainable, scalable code.
- Built distributed systems using UDP + Flatbuffers and asynchronous queues for inter-process communication, ensuring robust synchronization of RL training steps with simulation states.
- Improved developer workflows by integrating third-party dependencies, refining CMake-based builds, and containerizing development environments for efficient CI/CD and cross-platform compatibility.
- Expanded simulation capabilities with C++ plugins for AFSIM, introducing custom instrumentation of scenario data and specialized sensor modeling to better inform RL policies without altering core simulation fidelity.

**Software Engineering Intern**, Torch Technologies – Huntsville, AL 05/2020 – 08/2022  
(Summers/Winters)

- Created C++ and Python integrations with an internal microservices "Toolbox" and AFSIM simulation pipelines, generating large-scale training datasets under distributed, logically clocked conditions.
- Developed a Python-based ML pipeline to preprocess simulation-derived data, applying various algorithms (GBM, KNN, DNN) and visualizing performance metrics in Grafana to inform model selection.
- Designed a Python GUI (Terrain Tool) for manipulating high-resolution elevation and imagery data, automating the segmentation and export of terrain tiles into Unreal Engine environments.
- Enhanced Unreal Engine C++ front-end visualization by incorporating new DIS packet handling for dynamic

scenario elements, as well as implementing a deterministic network replay tool for reproducible debugging and performance analysis.

**Undergraduate Research Assistant**, Mississippi State University CSE Department

09/2020 – 05/2022

- Developed data acquisition/analysis systems for predictive maintenance using a machine fault simulator and fiber optic sensors.
- Researched transfer learning and tested advanced ML clustering algorithms (TICC).
- Built and studied GANs for high-quality dataset expansion.

## SKILLS

---

- Expert in C++ OO design (factory, strategy, abstraction), data structures, memory management, and multithreading.
- Proficient in distributed systems, IPC mechanisms (ROS2 pub/sub, UDP + Flatbuffers), and synchronization techniques.
- Skilled in building and maintaining complex codebases with CMake, Python bindings, and containerized CI/CD workflows.
- Experienced in integrating ML/DL frameworks (RLlib, SB3) with simulation environments, optimizing training pipelines and inference performance.
- Adept at embedded/systems-level concepts, logical time simulation, and cross-platform development on Linux/Windows.

## PROJECTS

---

### Autonomous RC Car (Minesweeper):

- Solely engineered the entire software stack for a low-cost, disposable autonomous RC car designed to sweep for and detonate landmines.
- Architected a robust ROS 2 pub/sub service-oriented navigation system fusing raw IMU/GPS data for precise Ackermann steering control via electronic speed controllers.
- Implemented a Python-based command controller integrating base station input and RTK correctional data over a mavros radio, ensuring accurate real-time navigation.
- Developed all supporting base station software and communication interfaces, seamlessly linking computational logic with field operations.

### Tiger Language Compiler:

- Designed and implemented a C++ compiler for a functional, ALGOL-like language (Tiger), producing optimized MIPS assembly.
- Introduced a TAC intermediate representation and implemented two register allocation strategies:
  - Chaitin's Naive Allocation: A simpler, less optimized approach relying on heuristic allocation.
  - Briggs Optimization: A comprehensive scheme incorporating control-flow and liveness analysis, as well as instruction cost metrics to minimize load/stores and branches.
- Achieved best-in-class performance on benchmark tasks, reducing instruction counts by a factor of 10 compared to peer solutions.

### Before and After Creator:

- Built a locally hosted web application leveraging unsupervised ML and deep learning to match before/after car detailing images.

- Automated bulk image processing, generating side-by-side collages for a detailing business, improving workflow efficiency and consistency.
- Engineered a custom pairing algorithm to reliably identify corresponding images and streamline final outputs for user download.