

PATRICK LIU

patrick.liu.422@gmail.com | +1(314)760-8355 | St. Louis, MO | [linkedin.com/in/patrick-liu-2914712ba](https://www.linkedin.com/in/patrick-liu-2914712ba)

EDUCATION

Washington University in St. Louis

B.S. in Computer Science; Second Major in Mathematics

St. Louis, MO

Aug 2023 – May 2027

- GPA: 3.97/4.00
- Relevant Coursework: Object-Oriented Software Development, Artificial Intelligence, Machine Learning and Pattern Classification, Data Structures and Algorithms, Data Science, Computer Engineering, Parallel and Concurrent Programming, Number Theory and Cryptography

EXPERIENCE

Pipeworks Studios

Eugene, OR

Software Engineering Intern

Jul 2026 – Aug 2026

- Developed backend gameplay systems and logic for an unreleased Disney cross-platform multiplayer card game using **C#** and **Unity**, collaborating within a 20+ person cross-functional team on an existing production codebase.
- Investigated and resolved complex card-targeting and selection logic by tracing issues across **C# scripts and the CLIPS rule-based system**, correcting interactions between card-specific mechanics and shared game rules.
- Implemented and debugged card keyword mechanics by translating game design specifications into production gameplay logic, resolving **20+ engineering tickets within the first three weeks**.
- Debugged edge-case interactions between individual card mechanics and shared game rules, implementing targeted fixes while preserving expected behavior across related cards and legacy mechanics.
- Tested fixes across edge cases and wrote corresponding test cases to validate gameplay behavior and prevent regressions across related card interactions and shared systems.
- Integrated production code changes through team code review and CI/build workflows using **Perforce and Jenkins**.

Virtuos

Prague, Czech Republic

Software Engineering Intern

May 2025 – Jul 2025

- Delivered 2 playable **Unreal Engine 5** mini-game prototypes by implementing core gameplay logic and systems (actors/components, event-driven interactions) using **Blueprints and C++ scripting**.
- Rapidly ramped up on UE5 and Blueprint workflows within the first two weeks, iterating on gameplay features toward demo-ready builds under a short development timeline.
- Debugged gameplay systems using breakpoints and logs to identify root causes, resolve logic defects, and improve stability and responsiveness through iterative playtesting in Unreal Editor.
- Developed and iterated on features in an Agile/Scrum environment, incorporating peer feedback through sprint planning, daily standups, demos, and Git-based version control.

Handshake AI Fellowship

San Francisco, CA

AI Trainer & Independent Research Contributor

Aug 2025 – Sep 2025

- Improved training signal quality for **LLMs** by producing and evaluating high-quality prompts, responses, and reasoning traces aligned to factuality and instruction-following goals.
- Evaluated LLM prompts and responses for factual accuracy, instruction following, and reasoning quality, providing structured feedback to improve training data quality.
- Improved evaluation consistency by applying structured verification (task completion checks, pass/fail criteria) and strictly adhering to verification heuristics and providing evidence-based rationales for edge-case pass/fail judgments.
- Strengthened model reliability efforts through iterative error analysis: identifying failure modes (hallucinations, unclear instructions, policy non-compliance) and providing targeted, actionable feedback.

PROJECT EXPERIENCE

Extensible File System Simulator (C++)

Jan 2025 - May 2025

- Designed and implemented a **modular file system simulator and CLI in C++**, supporting file creation, metadata inspection, copy, rename, removal, and overwrite/append operations.
- Increased extensibility and reduced coupling using **Command and Factory patterns**, separating command execution from file object creation.
- Applied the **Visitor Pattern** to separate file system operations such as rendering and size calculation from the directory structure, enabling new functionality without modifying existing classes.
- Integrated access control for protected files using the **Proxy Pattern**, preventing unauthorized deep-copy and write operations.

SKILLS

- **Languages:** Python, JavaScript, TypeScript, Java, C++, C#, SQL, Assembly (x86 / AVR), HTML5, CSS3, LaTeX
- **Tools/Tech:** Git, GitLab, Perforce, Docker, CMake, React, Node.js, REST APIs, NumPy, Pandas, scikit-learn, Matplotlib, Linux/Unix, Unreal Engine 5, Unity, Jenkins, AWS (EC2, S3)