

# Hyeonjoon Nam

Software Engineer

hyeonjoon.nam.dev@gmail.com | (206) 786-7888 | [linkedin.com/in/hyeonjoon-nam](https://www.linkedin.com/in/hyeonjoon-nam) | [portfolio-sooty-rho-46.vercel.app](https://portfolio-sooty-rho-46.vercel.app)

## SUMMARY

- Software engineer focused on C++ for games, with projects in Unreal Engine, Unity, and custom OpenGL-based C++ engines. Experienced implementing AI behaviors, pathfinding (including A\* optimizations that cut CPU time from 8.4s to 0.65ms), and server-authoritative combat, and I like building gameplay systems and tools that are easy to debug, measure, and iterate on.

## SKILLS

**Languages:** C++, C, C#, Python, JavaScript

**Engines & Graphics:** Unreal Engine, Unity, Custom C++ 2D engines (OpenGL-based)

**Tools / Platforms:** Perforce (Helix Core), Git, GitHub, MySQL, Visual Studio, VS Code

## EXPERIENCE

### QA Lead / Gameplay Programmer

09/2025 – Present

Kimhap Games - Triad of Valor | 3D MORPG game | Unreal Engine, Perforce

- Implemented a **server-authoritative melee combat system** by driving hit checks from Anim Notify States on the attack montage, performing overlap checks only on the server and multicasting the results to clients.
- Integrated the custom **p4bot Perforce→Discord automation** tool into the team pipeline, giving artists and designers real-time visibility into file commits and checkouts and **significantly reducing accidental content conflicts**.

### Tech Lead / Gameplay Programmer

09/2024 – 04/2025

Derpy Doggo Digital - The Children Are Sleeping | 3D horror puzzle game | Unreal Engine, Perforce

- Implemented the monster AI states (Patrol, Chase, Catch, Prediction), replacing naive random navmesh sampling with point-based routes and a weighted patrol system. In internal tests on one of our main levels this brought first-encounter times into a **12–48s** window (previously **30–100s**) and stabilized encounters at around 2 per 3–5 minute run.
- Led a 4-person programming team in a 13-person project, turning pacing and tension goals from design into concrete engineering tasks and using weekly cross-discipline meetings to negotiate scope and keep implementation on track.

### Tech Lead / Gameplay Programmer

09/2023 – 12/2023

Exodia - They Are | 2D defense game | Custom C++ 2D engine, Git/GitHub

- Designed and implemented the core combat loop, including tower placement, deletion, and upgrade rules, monster wave spawning, and enemy path selection toward the base.
- Optimized enemy pathfinding by computing a single shared path for all enemies whenever the map changed, reducing A\* calls from 24,000 to 1 in a 100-enemy benchmark and cutting total pathfinding CPU time from **8.4s to 0.65ms** while keeping paths identical.

### Tech Lead / Gameplay Programmer

03/2024 – 06/2024

Exodia - Bastion | 2D defense game | Custom C++ 2D engine, Git/GitHub

- After the project was rebooted between semesters, analyzed the previous codebase, salvaged reusable systems, and reorganized the core game mechanics so the team could ship a new version within the remaining term.
- Built a data-driven loading layer that parses wave configurations from external text files and instantiates units at runtime, cutting the end-to-end “save, build, and reload the level” loop from **62 seconds to 15–17 seconds (a 4× improvement)** and making it much easier to retune wave timing and composition without touching C++ code.

### Tools / Automation Developer

09/2025 – Present

Personal Project - p4bot | Self-hosted Perforce → Discord automation | PowerShell, Python, Perforce

- Implemented a self-hosted bot that periodically runs Perforce command-line queries to list recent changelists and opened files, then sends structured Discord embed messages so the team can see who changed what at a glance.
- Summarizes depot state (adds/edits/submits/reverts) and exposes a /canwork command teammates use to check whether a file is currently in use or safe to modify, all running continuously in the background on a local Windows machine.
- Cut Perforce lock-check overhead from roughly **70–90 seconds** (and sometimes several minutes) to a **1–3 second** Discord glance, saving our 6-person team about **10–20 minutes per day** — roughly 4–10 hours over the term.

## EDUCATION

### B.S. in Computer Science in Real-Time Interactive Simulation

DigiPen Institute of Technology

Expected 04/2026