

Gameplay and tools programmer working in **Unreal Engine since 2021**. I build gameplay systems, then the editor tools that let designers use them without me: a quest graph editor, a talent tree, a developer console and an actor pool with its own debugger, each written to unblock a real team.

SELECTED WORK

Slime Slinger / Gameplay & tools programmer 2024 - 2025

Team of 9 · 9 months · UE5, C++, Blueprints · PC / Steam Deck

- Owned the **ability and combat frameworks** for a nine-person team, built on composed components rather than an inheritance hierarchy so designers could add behaviour without a programmer.
- Moved every reusable system out of the game module into **24 in-house plugins**; four now run in other projects unmodified.
- Cut a level's texture memory from **1.3 GiB to 1.0 GiB**, then found a Blueprint cast making **every enemy load the entire player character**: 137 MiB per AI, fixed with an interface call.

Frog Wood / Sole programmer 2023

Team of 7 · 6 months · UE5, C++, Blueprints · PC

- Delivered **every gameplay, UI and technical system** on a 3D platformer as the only programmer, and owned version control and **build stability** for the team throughout.
- Built **Game Actions**, a composable behaviour library that let designers author doors, cinematics and objective updates without engineering support.

Nerve Quest / Tools programmer · UE5 plugin 2025

Solo · 2 months · C++, Slate, Blueprints

- Replaced hard-coded quest logic with a **graph editor that compiles to flat runtime data**, so designers author branching quests, including nested sub-quests, with no code per quest.
- Validation runs **as connections are made**, so a wire that would strand an objective is refused while the designer is looking at it rather than found by QA a fortnight later.

Dynamic Object Pool / Performance engineer · UE5 plugin 2025

Solo · 2 weeks · C++, Blueprints

- Diagnosed a combat scenario dropping **60 FPS to 15** (the cost was actor spawning, not rendering) and restored a stable 60 with a thread-safe generic pool, cutting allocations by roughly 85%.
- Adoption only happened once I added **custom Blueprint nodes**; the C++ API alone went unused for two weeks because designers had a SpawnActor node and no reason to change.

Developer Console / Dev tools engineer · UE5 plugin 2025

Solo · 3 weeks · C++, UMG, Blueprints

- Took a balance tweak from a **half-hour recompile cycle down to about thirty seconds**, and let QA test god mode, talent grants and checkpoint teleports without pulling a programmer off their work.
- Commands declare their parameter types and the console **generates its own UMG interface** by reflection, so a new command needs no widget wiring. Stripped entirely from shipping builds.

Nerve Talent Tree / Tools programmer · UE5 plugin 2024

Solo · 1 month · C++, Slate, Blueprints

- Node-based progression editor whose schema makes an **invalid tree unauthorable**: circular prerequisites are refused at edit time instead of deadlocking at runtime. Compiles to cached lookups, so a talent check is a map lookup, not a graph traversal.

SKILLS

Unreal Engine 5	C++ · Blueprints · Slate · editor modules · graph tooling · UMG · GAS concepts
Systems	gameplay & combat frameworks · component architecture · editor tooling · save / persistence
Performance	profiling · memory & asset references · pooling · async loading · thread safety
Unity	C# · prototyping · 2D / 3D · physics
Process	Git · Perforce · Jira · Scrum · build stability · teams of 5 to 9

EDUCATION & ALSO

Falmouth University / BA (Hons) Game Development: Programming 2022 - 2025

Miquella's End, solo boss-encounter study (2023) · Crab Tank, Global Game Jam 2025 · The Organic Route, 72-hour jam (2022). Write-ups on the portfolio.