

Liam Suter

+1 647-996-9255 | liam.suter0@gmail.com | linkedin.com/in/liam-suter | liamsuter.com

Education

University of Waterloo

Sept. 2024 – May 2028

Bachelor of Computer Science w/ Co-op (GPA: 3.96/4.0)

Waterloo, ON

Technical Skills

Languages: C++, C#, C, Python, Java, Rust

Game Development: Unity (URP), Unreal Engine 5, Networking & Replication, HLSL, Shader Graph, Niagara VFX, Level Design

Tools & Practices: Git, Perforce, Unity Profiler, Visual Studio, Jira, Confluence, Trello, Grafana

Experience

Gameplay Programmer | *Tactic Studios*

Jan. – Apr. 2026, Sept. 2026 – Present

- Spearheaded the design and implementation of **Resurrection**, a core mechanic spanning **5 gameplay systems**, which shipped as the primary feature of a major update, adding depth and strategy to the core loop
- Cut packet sizes of high-frequency **net messages** by **15%**, reducing server-client bandwidth and latency
- Implemented Intermissions, a **major feature** adding new strategic and social depth to matches
- Diagnosed and resolved critical **microtransaction bugs** in a live game, protecting revenue and ensuring a reliable purchase flow for players
- Generalized a single-context **event-dispatch system** into a context-container model, supporting complex events without changing the dispatch interface
- Fixed **20+ client-server desync** bugs caused by relevancy scoping and inconsistently handled net messages

Unity Game Developer | *Original Studios*

May 2025 – Aug. 2025

- Designed and implemented an injury system in C#, integrating gameplay logic, UI, and event handlers, which shipped as the core feature of a major update that boosted concurrent players by **over 80%**
- Developed **multipass HLSL screen-shaders** in Unity's **Universal Render Pipeline** to visually convey negative status effects, strengthening player feedback and improving readability
- Optimized resource spawning logic, cutting its per-frame cost by **75%** while enabling more natural spawn patterns
- Built a **save/load system** that detects and self-corrects invalid states, protecting progress from data corruption

UE5 Gameplay Programmer | *Curiouser and Curiouser Games (Indie team, 5 devs)* Apr. 2024 – Aug. 2026

- Introduced a **dynamic map-generation** system in C++ that randomizes item and key-location spawns with distance-based weighting for fair, varied layouts
- Implemented client-server networking with **RPCs** to synchronize key game events across all players
- Expanded replayability with a **dynamic quest system** that assigns players varied tasks each round

Projects



Seas of Static | *Unity, C# Scripting, Data-Driven Design, Cinematics, Level Tooling*

May 2025 – Present

Strategy/puzzle game set in a war-torn world, with a playable demo on Steam

- Designed a modular **cinematic builder** that chains cinematic and gameplay components into single sequences, letting story moments play out without pulling the player out of the game
- Created a **data-driven condition system** that sequences gameplay events, describing complex situations through small composable condition blocks
- Developed a **JSON-based level builder** with per-mission control over enemies, weather, and modifiers, enabling fast mission creation and iteration
- Implemented a keypoint-driven **camera system** for smooth movement and framing inside the cabin



Dune Drifter | *Unity, C# Scripting, Enemy AI, Gameplay Systems*

Oct. 2026

Endless desert shooter built in 72 hours with a team of 6 for the UW Fall 2026 Game Jam

- Designed the **enemy AI** behaviour system for **6 unique enemy types** with distinct pursuit and attack patterns
- Created a **Spawn Director** that coordinates enemy spawns to pace encounters and create more impactful gameplay moments
- Built a dynamic **weapon-swapping system** that lets players shape their own playstyle
- Created a custom simplified **vehicle controller** and **physics system** for enemy units