Aman Singh

Amansinghdev.com || asa259@sfu.ca || Showcase Reel

SUMMARY OF QUALIFICATION

- Gameplay Engineer specializing in **C++ and C#** gameplay system development across **Unreal Engine** and **Unity**. Focused on **FSM-driven AI**, **hitscan combat pipelines**, and **physics-based traversal systems**, building modular and performance-conscious architectures grounded in **3D vector math** and frame-rate-independent design — from engine-level movement components to scalable encounter systems.
- **Languages:** C++, C# | **Game Engines:** Unreal Engine, Unity | **Tools:** Visual Studio, Git, Maya

EDUCATION

Bachelor of Arts, Double Minor, Computing Science & Interactive Arts and Technology Sept 2019 – Dec 2024

Simon Fraser University, Burnaby, BC

Cumulative GPA: 3.03; Dean's Honor Roll 2023 (Spring)

TECHNICAL PROJECT EXPERIENCE

Advanced FPS Jetpacking Movement (Unreal Engine, Blueprints + C++) Oct 2025 – Present

Gameplay Programmer

- Designed a modular **multi-state locomotion architecture** (Walk, Sprint, Crouch, Slide, WallRun, Jetpack, Prone, Airborne) with centralized state transitions and deterministic **conflict resolution** between overlapping movement systems.
- Implemented an analog, **curve-driven jetpack system** with frame-rate-independent fuel logic using **timer-based updates**; identified and resolved a **timer synchronization bug** affecting short-thrust fuel drain behavior.
- Developed a custom **C++ async root motion wrapper** (UCancellableAsyncAction) exposing cancellable, curve-driven forces to Blueprints with **OnComplete/OnFail delegates** and RootMotionSourceID lifecycle management — replacing **LaunchCharacter** impulses with predictable, tunable root motion forces.
- Integrated an FPS animation pack via **interface-driven component access** (CameraAnimator, GAIT locomotion), enabling seamless viewmodel transitions across all locomotion states with layered **ADS, recoil, and FOV** effects.
- Migrated UE4 ShooterGame template to UE5 via **C++**, resolving deprecated APIs and modernizing the rendering pipeline (**Lumen, VSM, TSR, DX12**) for a playable demo environment.

The Storge — Horror 2D Game (Unity, C#) Sept 2023 – Dec 2023

Gameplay Programmer

- Architected a top-down controller system with **animation blending**, Rigidbody2D-based movement, and synchronized **animation-event-driven damage registration** for accurate hit timing.
- Implemented a **multi-state AI framework** (3+ behavioral states) combining **A* pathfinding**, proximity detection, and dynamic component activation (**AIPath enable/disable**) to control chase pacing and tension.
- Developed noise-driven and **line-of-sight enemy detection** systems, dynamically rerouting pathfinding targets and modifying aggression states based on environmental triggers.
- Created a custom **Unity Editor reflection tool** to automatically convert tilemap geometry into **ShadowCaster2D** components — eliminating manual shadow placement and reducing lighting iteration time.
- Designed a sound-driven stealth loop using **3D positional AudioSource** emitters in a 2D game — wiring trigger-lifetime noise objects to enemy patrol retargeting to reinforce psychological tension and player feedback.

Slow & Steady — Rogue-like 2D Shooter (Unity, C#) Sept 2023 – Jan 2024

Gameplay Programmer

- Engineered a modular **hitscan combat system** with cooldown logic, tracer VFX, and decoupled camera feedback — separating input (**Update**) from physics (**FixedUpdate**) to ensure frame-rate-independent gameplay, achieving stable **80–120 FPS**.
- Implemented a time-scaling **slow-motion mechanic** by dynamically adjusting **Time.timeScale** and compensating physics forces and camera interpolation — preserving responsive player control under altered temporal states.
- Developed a multi-archetype **enemy AI suite** (4+ enemy types + boss) using **FSM architecture** and **A* pathfinding**, supporting patrol, chase, attack, AoE behaviors, and coroutine-driven multi-phase logic.
- Designed a scalable **wave-spawning system** managing randomized spawn points, enemy queues, and wave completion tracking — enabling controlled difficulty ramping across 5+ waves.

Paradox Rift — VR Exploration Game (Unity, C#) Jan 2024 – April 2024

Gameplay Programmer

- Implemented XR locomotion systems including teleportation, smooth movement, and two-handed grab mechanics using **Unity XR Interaction Toolkit** — supporting 6DOF headset tracking and real-time hand pose animation.
- Designed an **interface-driven modular AI architecture** enabling patrol, tracking, and attack states with **NavMesh navigation** and custom raycast-based **line-of-sight logic**.
- Built scalable interaction systems using **UnityEvents** and component-driven design, allowing reusable interactive objects across 5+ gameplay scenarios.
- Optimized physics-based interactions by tuning **collision layers**, rigidbody interpolation, and interaction thresholds to maintain stable **70–90 FPS** VR performance targets.