Aman Khandelwal — Resume (Remote, 2026)

Nov 8, 2026

Senior Gameplay & Systems Programmer — Unreal Engine 5 / Unity · Open to remote

Jaipur, India (IST, UTC+5:30) · khandelwalaman5@gmail.com · +91 8619067110

GitHub · ArtStation · Fab · YouTube · LinkedIn

Summary

Senior Gameplay & Systems Programmer with 5+ years shipping games across PC, mobile and VR in Unreal Engine 5 and Unity. Specializes in C++ gameplay architecture, server-authoritative multiplayer systems, AI frameworks, custom editor tooling and performance optimization. Shipped titles on Steam, Google Play, the App Store and the Meta Quest Store.

Independently designed, built and commercialized seven products on Fab — reusable UE5 C++ frameworks, a Unity AI suite, gameplay tooling and a 750+ asset low-poly environment series — owning each end to end from architecture through documentation, packaging and publishing. Comfortable working autonomously in distributed, remote teams.

Technical Skills

Languages	C++ (UE5 gameplay framework, replication, editor modules), C#, Python
Unreal Engine 5	C++ modules & plugins, Blueprints, Gameplay Tags, GAS, custom editor tooling (asset editors, detail customizations, validators), data assets, animation blueprints & state machines, MetaHuman, Cinematics
Unity	URP, Burst, WebGL & mobile builds, AR Foundation, GraphView-based custom editors, Behavior/AI tooling
Multiplayer & Networking	Server-authoritative simulation, actor replication, dedicated servers (AWS), Photon Fusion, Epic Online Services, Steam SDK, matchmaking, voice/text chat
AI & Gameplay Systems	Behavior Trees, Environment Query System (EQS), Perception, Smart Objects, crowd simulation, NPC AI
Software Architecture	Reusable framework design, Blueprint-extensible APIs, automated testing (EditMode/PlayMode), modular plugin packaging
Backend	Firebase (Auth, Firestore, Cloud Functions), Google Play Services, AdMob
Content Pipeline	Blender (bpy) procedural generation, cross-engine (Unity/UE5) asset packaging
Profiling	RenderDoc, Unreal Insights / Session Frontend, GPU Visualizer
Tooling & Publishing	Git, Perforce, Visual Studio, Xcode, Fab / ArtStation product publishing
AR/XR	Meta Quest, Oculus SDK, MapBox SDK, location-based AR
Languages Spoken	English (professional working proficiency), Hindi (native)

Selected Engineering Work

HITZ: Merge Tactics — UE5 Auto-Battler Framework (solo, published on Fab)

Unreal Engine 5.8 · ~13,800 lines of C++ across four modules · Paid, source-available listing on Fab · Fab · Code · Docs

Built for UE5 developers who want a complete, playable auto-battler / merge-tactics game out of the box, with the option to extend it without touching C++.

- Designed a reusable, source-available framework — not just a feature set — that lets another UE5 developer ship a complete auto-battler without writing C++. Wrote the listing, the setup guide and the full documentation site myself.
- Split it into three reusable modules (`HMT_Core` 6.2k lines, `HMT_CoreEditor` 1.3k, `HMT_GASBridge` 0.3k) plus a separate sample-game module (6.0k). The framework drops into an existing project by copying three source folders.
- Built server-authoritative combat. The server locks board snapshots, simulates, and replicates a compact event log that clients replay for animation. Clients never compute outcomes, which closes the cheat surface on match results.
- Designed the extension surface around four Blueprint-implementable interfaces (`IHMT_StatProvider`, `IHMT_TargetingStrategy`, `IHMT_TraitEffect`, `IHMT_AbilityExecutor`), so buyers ship a complete game without writing C++.
- Wrote a GAS bridge that maps the stat interface onto an existing `AttributeSet`. GAS-based projects swap the stat backend per unit with no changes to any other system.
- Built editor tooling: a click-to-paint board layout editor, a Unit Definition asset editor with live 3D preview and per-montage playback, and a save-time validator covering seven data-asset types.
- Implemented the full multiplayer round loop on standard UE replication: preparation, combat and results phases, recency-weighted pairing, ghost boards for odd lobbies, PvE rounds, spectating and late-join.
- Supports hex and square grids from one board asset, with neighbour, distance and transform math decoupled from board state.

Hitz AI Suite — Unity AI Framework (solo, published on Fab)

Unity 6000.3 LTS+ · C# · nine interoperable AI systems plus a Unity Behavior bridge · 337 automated tests (271 EditMode + 66 PlayMode) · Fab

- Designed and built a commercial Unity package that brings Unreal-style AI tooling — Behavior Trees, Environment Query System (EQS), Perception, Smart Objects and crowd simulation — into Unity, including a bridge into Unity's own Behavior package.
- Built a GraphView-based Behavior Tree editor with auto-layout, stable sibling sort and save-time key validation.
- Shipped with an integrated showcase demo: an Outpost sample, a scripted Boss Fight encounter, an S500 stress-test scene and 11 per-system samples, all code-built and mirrored into the package via a Python sync script.
- Performance-tuned for production: a 500-agent crowd-simulation benchmark (S500) holds a 0.19 ms median per-frame cost with zero GC allocation, measured in the Unity Editor.
- Covered by 337 automated tests (271 EditMode, 66 PlayMode); release builds strip all debug/editor-only tooling via conditional compilation.
- Added project/level/per-platform AI settings with live apply, so buyers tune frame budget without touching code.

Ktrl Smash — Multiplayer Arena Brawler

Unity · Steam (Windows) + WebGL · Released Dec 2025 · Mostly Positive on Steam

- Shipped a free-to-play online PvP brawler. Architected and implemented multiplayer gameplay using Photon Fusion, collaborating with DevOps to deploy and operate dedicated game servers on AWS.
- Fixed a latency climb over the length of a session (20→60 ms on Windows, 40→100 ms on WebGL). The cause was per-frame work piling up on WebGL's single main thread, made worse by frequent network updates and redundant replicated state.
- Audited coroutine usage across the codebase, cut update frequency and payload size, and removed unneeded replicated variables. Ping now stays flat for the whole session.

The Cursed Hour — 4-Player Co-op Survival Horror

Unreal Engine 5 · Steam · Released Nov 2024 · 129 reviews, Mostly Positive · ~18–23k estimated players (estimated from Steam review count via the standard reviews-to-owners ratio)

- Pitched the original concept and owned the core multiplayer systems through to ship.
- Developed the core multiplayer architecture, including server-connected ghost AI, lobby/session management, and integrated text and voice communication.
- Integrated Epic Online Services and the Steam SDK in parallel for a planned cross-platform rollout, keeping the session layer backend-agnostic.

DinoMap Explorer — Location-Based AR Collection Game (client project)

Unity · Android & iOS · In development · Team Lead

- Lead developer on a location-based AR game where players find and collect dinosaurs in the real world, with an educational layer.
- Built the AR capture flow, the in-game dinosaur encyclopedia and NFT integration. Several of these systems became shared components reused across other studio titles.

Experience

Senior Game Developer — Fiftyfive Gaming Studio Pvt Ltd

Oct 2021 – Present · Jaipur, India · Teams of 5–6 (2 devs, modeller, animator, UI designer)

Gameplay, combat and systems programming on shipped titles for PC (Steam), mobile (iOS/Android) and VR (Meta Quest).

- **Rift Riders** (UE, iOS/Android): proposed the concept, then built endless track generation, the vehicle pawn, quest and task systems, and Firebase/Play Services integration.
- **Fight Hero** (Unity, iOS/Android): owned the mech combat loop, enemy waves, and upgrade/progression balancing. Directed the art team on combat animation and gameplay assets.
- **Combat animation packs** (Fab, ArtStation): led the product from idea to store, ~$6,000 revenue (see Independent Products & Commercial Publishing).
- Fixed a persistent iOS `bad_weak_ptr` crash caused by Firebase linking twice: Unity prebuilt libraries alongside EDM4U's Swift Package Manager resolution.
- **Tower of Demons** (Unity): added game modes, cosmetics, resolution-scalable UI and Firebase-backed player data.
- **Nutrients Sold Separately** (UE, Play Store): Epic game jam entry, team of 5, "As Above, So Below" theme.

Unreal Game Developer (Intern) — Hyperreality Technologies

Jun 2021 – Oct 2021

- **Robo XR** (UE, Meta Quest): sole developer on a robot build-and-battle VR game shipped to the Meta Quest Store. Built the robot assembly system, combat, enemy AI and DLC support.

Independent Products & Commercial Publishing

I design, build and sell commercial Unreal Engine and Unity products on Fab, Epic Games' marketplace, plus animation packs on ArtStation. Full engineering detail on the two flagship frameworks is under Selected Engineering Work above.

- **HITZ: Merge Tactics** (Fab · Code · Docs) — UE5.8 auto-battler framework, source-available paid plugin. See Selected Engineering Work.
- **Hitz AI Suite** (Fab) — Unity AI framework (Behavior Trees, EQS, Perception, crowd simulation, Smart Objects). See Selected Engineering Work.
- **Hitz GeoMap** (Fab · Product site) — Blueprint-only UE5 plugin that streams real-world map geometry (roads, buildings, water, parks) around the player's GPS position from five vector-tile providers (OpenStreetMap, Mapbox, MapTiler, Stadia Maps, VersaTiles) or a self-hosted server. 13 styleable layers, rule-based mesh placement (trees, road-side props), deterministic actor spawning on real points of interest, live GPS/compass input on Android. Windows + Android.
- **HitzGAS Toolkit** (Fab · Product site) — Precompiled UE5 plugin that drops the Gameplay Ability System onto a Blueprint actor via three components (attribute init, attribute-change listener, Enhanced Input-to-ability mapping), with a DataTable-driven attribute set, six sample abilities/effects, a live GAS debugger (attributes, abilities, active effects, exportable to JSON), and a C++ AttributeSet generator.
- **Combat animation packs** (ArtStation) — spotted the product opportunity and directed the artist team; built the UE integration layer myself (animation blueprints, state machines, root-motion and in-place variants) plus packaging and publishing. ~$6,000 in revenue over two years, still earning.
- **Cozy Kingdom series** — Cozy Kingdom Builder Kit and Cozy Christmas Village (Cozy Halloween Village pending Fab approval) — 750+ hand-authored low-poly assets across Unity and UE5, built with a custom Blender/Python procedural generation pipeline and a demo layout shared across both engines' sample scenes.
- Own the full product cycle on every listing: scoping, building, documentation, store pages and publishing.

Education

B.Tech, Computer Science (Cloud Computing specialization), Poornima University, Jaipur · 2018–2022