Leon Yun

VR Engineer / Game Programmer

Surrey, BC · (416) 602-1525 · cheukho.leon.yun@gmail.com · leonsnotes.ca · github.com/cheukhoyun

PROFESSIONAL SUMMARY

VR engineer with three years shipping commercial VR at LuckyVR — the PSVR2 port and PS5 Non-VR port of *PokerStars VR / Vegas Infinite*. Deeply in love with the craft of game and VR development: currently writing *VR 101: Foundations of VR Interaction*, an in-progress book on VR rig architecture, hand interaction, and locomotion. Recently built an experimental sphere-world engine where the planet's surface is treated as intrinsic flat geometry rather than an embedded mesh. Works across gameplay systems, tooling, VR interaction architecture, and physics simulation.

PROFESSIONAL EXPERIENCE

Software Engineer & Technical Partner

Jun 2025 – Present

Private Contract

- Built and shipped a desktop product (Electron + React client; TypeScript / Express + Prisma / PostgreSQL backend), running in production on PM2 with structured logging and request tracing.
- Architected a real-time task execution engine: singleton task manager driving concurrent background work via a 100 ms tick loop, with full state recovery from the database on restart.
- Designed a hash-based client-server sync protocol for efficient delta updates of complex user / token state, with matched hashing on both ends and no optimistic frontend writes.
- Built a multi-process Electron architecture with MITM proxy child processes, IPC orchestration across main / renderer / workers, self-signed CA certificate handling, and Windows PAC-based proxy routing.
- Reverse-engineered a third-party API's signature and encryption scheme, integrated through a serverless proxy layer for IP rotation and rate-limit resilience.

Software Engineer / Game Programmer

Aug 2022 – Jun 2025

LuckyVR Inc. — Toronto, ON

- Shipped the PSVR2 port and PS5 Non-VR port of *PokerStars VR / Vegas Infinite* on a small PlayStation team, contributing across systems, VR mechanics, gameplay, and tooling.
- Replaced Meta-based avatars with Ready Player Me, re-integrating full-body IK, hand pose / grab systems, throwing physics, locomotion, seated and standing pose handling, and per-user calibration so the new avatars dropped into the existing VR interaction stack without regressions.
- Designed and integrated experimental hardware-facing systems (eye tracking, adaptive triggers, lip-sync) and built in-engine tools that let designers tune haptics, animation, and interaction behavior without a programmer in the loop.
- Adapted VR gameplay systems for traditional gamepad control on the Non-VR port — redesigned input mappings, interaction logic, and camera handling so the game felt native on a gamepad without re-teaching VR players.
- Worked across gameplay flow: multiplayer state synchronization, table / seat management, player onboarding, and cross-platform feature parity between VR and Non-VR builds as features shipped.
- Focused on platform-specific bug fixing for the PSVR2 port — chased down crashes, regressions, and certification blockers to bring the title through Sony's review.

Software QA Engineer (PEY Co-op) *Sep 2020 – Sep 2021*

GEO Semiconductor — Toronto, ON

- Manual and automated chip-functionality testing; built Python / C++ scripts extending the team's ROBOT Framework modules to expand test coverage.

Private Programming Tutor *2019 – Present*

Self-Employed

- Provided 1,000+ hours of tutoring to university students in Python, C / C++, Haskell, and Lisp-family languages.

SELECTED PUBLICATIONS & PROJECTS

VR 101: Foundations of VR Interaction — *book in progress*

Technical book on VR interaction design: rig architecture (ghost / pawn / avatar separation, recenter, pursuit), locomotion, hand interaction, avatar IK, and immersive UI/HUD. Drafts and table of contents at leonsnotes.ca.

Planet Engine — A 3D Engine Where the Sphere *Is* the World

Built an experimental engine where a planet's surface is treated as intrinsic flat geometry on a sphere ($S^2 \times \mathbb{R}$), not as an embedded mesh. Players walk on perfectly flat ground that nonetheless wraps around — flat physics, flat pathfinding, no embedded-sphere tricks. Each entity's pose is stored as an SO(3) matrix to make pole-crossing a single matrix multiply rather than a coordinate-singularity case. Interactive demos and full writeup published at leonsnotes.ca.

EDUCATION

B.Sc. Computer Science, Specialist in Artificial Intelligence — University of Toronto. *Graduated with high distinction.* *2018 – 2022*

TECHNICAL SKILLS

Languages	C#, C / C++, Python, TypeScript / JavaScript, Haskell, Lisp-family languages, SQL
VR / XR SDKs	OpenXR, Meta XR SDK, PlayStation VR2 SDK, SteamVR, Unity XR Interaction Toolkit, Ready Player Me SDK
Engines	Unity (incl. DOTS / ECS) — primary; Unreal Engine, Godot — working familiarity; OpenGL
Web / Backend	Node.js, Express, Prisma, PostgreSQL, React, Electron, Vite
Infra / Tools	Git, PM2, Linux deployment, MITM proxying, ESP32 / embedded, automation scripting
Strengths	ECS architecture, tool development, technical writing & documentation, reverse engineering, cross-platform optimization