Sankeeth Ganeswaran

647-822-3351 | s2ganesw@uwaterloo.ca | LinkedIn | GitHub | Website

Education

University of Waterloo Sept. 2021 - April 2026
Bachelor of Computer Science, Co-op *Waterloo, ON*

Technical Skills

Languages: C#, C++, JavaScript, TypeScript, Python, Java, Kotlin, C, SQL, HTML/CSS, R, ActionScript 3
Frameworks: Unity, Unreal Engine, JavaFX, TensorFlow, PyTorch, Android, React, Flask, Swing, JUnit
Developer Tools: Git, Docker, Blender, Gradle, Postman, AWS, Oculus
Libraries: Pygame, NumPy, ThreeJS, Photon PUN, Unity VR, OpenAI Gym, JSoup

Experience

Unreal Engine Programmer May 2025 – Aug. 2025
Epic Games *Cary, NC (Remote)*

- Developed and optimized low-level engine and network replication systems in **Unreal Engine 5**, focusing on **Iris replication and dormancy**.
- Debugged and resolved complex engine-level issues, including **RPC behavior** and replication of **stable name subobjects**, to improve test reliability and network performance.
- Created **internal tooling** (e.g., replay rewrite tools) to streamline test automation and ensure backward compatibility.

Software Engineer Jan. 2025 – April 2025
X (formerly Twitter) *Palo Alto, CA*

- Spearheaded a **4-month initiative** to migrate Twitter/X's **2FA** from SMS and security keys to passkeys, impacting **250k+ active users** and designing the end-to-end user experience.
- Developed **Scala** backend services, **React** web flows, and **iOS (Swift)/Android (Kotlin)** mobile screens for seamless passkey enrollment and migration.
- Eliminated cross-domain redirects between **twitter.com** and **x.com**, reducing authentication requests by **50%**.

Game Developer Sept. 2023 – Aug 2024
ArenaX Labs *Toronto, ON*

- Developed features and machine learning systems for **AI Arena**, a platform fighter with **200k+** active players.
- Implemented core gameplay mechanics for the combat system in **JavaScript**, ranging from **finite state logic**, **animation systems**, **projectile physics**, **collision handling**, and implementing **20+** unique **elemental VFX** in **ThreeJS**.
- Resolved **50+** gameplay issues, revamped **raycast systems**, and cut down memory usage and load times by **40%**.
- Created a built-in **interactive tutorial**, implementing **15+** in-game demonstrations, with a **UI** designed with **React**.
- Worked on the AI agent for a final boss, designing a cohesive moveset and implementing **30+** state animations.
- Built several minigames in **Pygame** to train **reinforcement learning models** using **OpenAI Gym** environments.

Gameplay Programmer Jan. 2023 – April 2023
Lucky VR *Toronto, ON*

- Developed gameplay mechanics for the popular game **PokerStars VR** in **Unity**, for **PC, Quest, and PSVR**.
- Implemented scalable functionality in **C#** for **50+** new props and apparel for the **VR environment**.
- Fixed critical bugs and made significant QOL improvements for **70+** issues, ranging from **collision and physics interactions**, **backend integration**, **texturing**, and **networking** using **Photon PUN RPC** calls.
- Revamped several weapon systems and VR avatar interactions to improve performance and eliminate latency by **60%.**

Autonomous Vehicle Android Developer May 2022 – Aug. 2022
Ford Motor Company of Canada *Oakville, ON*

- Developed an **Android application** using **Kotlin** to send and receive **CAN and SOA messages** through the VHAL to set **60+** fundamental automobile property functions.
- Implemented an audio service for the infotainment system in Java using **Android Open Source Project Automotive** and **Google TTS service**, with the ability to play **40+** different prompts in **3** languages.

Projects

Creature Clash | *Unity, C#, Photon PUN*

- Developed a physics-based mobile game using **Unity** engine and exported to **Android** to publish to the Play Store.
- Used **Photon PUN** framework to create an online multiplayer lobby, with an **RPC** model.

MeetKicker | *JavaScript, HTML, CSS*

- Developed and published an extension to the **Chrome Web Store** that allows the user to kick themselves from a **Google Meet** once the members go below a customizable threshold, reaching **250+ active users** at its peak.
- Utilized **CSS** to design the extension panel and **Chrome API** to store user preferences in **browser storage**.