

Hello

I'm Ty (Timur) Radman

[Github](#) | [LinkedIn](#) | [Portfolio](#)

(604) 363 0155 | timurradman@gmail.com

Summary

Software engineer with 6 years of experience building solid, optimized, and maintainable gameplay systems in **Unity**, specializing in **C#** and **.NET**, and proficient in **Unreal Engine** and **C++**. I have shipped combat, AI, procedural generation, and multiplayer systems across large production codebases, and I love crafting fun solutions that enhance user experiences and simplify workflows for my teammates.

Experience

Jun 2025 - PRESENT

Graffiti+ (Tangible Interaction), Vancouver - *Software Developer*

Develop and optimize Unity-based interactive graffiti systems for live events across varied hardware.

- Supported 20+ live events for clients including **NASCAR**, **Formula 1**, **Samsung**, **Shopify**, and **Amazon**.
- Refactored the core codebase, reducing size ~30% and improving maintainability and runtime performance.
- Built Unity tooling (build debugger, printing framework, shader-based brushes, metadata-based asset packing with validation) to speed iteration and reduce build failures.
- Expanded the painting system: paint depletion with shake-to-recharge, neon glow, eraser, brush scaling by physical screen width, paint fade, and stroke preview improving accuracy for distance painting.
- Automated 3D model-to-paintable conversion, eliminating manual setup errors.
- Delivered gamified features such as wall-running entities and coverage-based scoring.
- Implemented generative-AI-based solutions in the app, including an AI stencil feature.

Jan 2025 - PRESENT

Retro Storm Studio, Vancouver - *Co-Founder*

- Plan and drive **Tank Lore**'s development cycle, defining milestones and scope through to its Steam release.
- Manage the studio website, set up and maintain the kanban board, and run weekly meetings to keep delivery on track.
- Links: [Steam Page](#)

Sep 2023 - May 2024

Centre for Digital Media, Vancouver - *Unreal Engine Developer*

- Developed core mechanics including the projectiles system and AI behaviors using **C++ Behavior Trees**.
- Implemented UI systems for the game's skills selection and equipment.
- Documented technical implementations for future development.
- Collaborated in an agile environment, participating in iterative development cycles and team reviews.
- Links: [GitHub](#) | [Project Link](#) | [Video Link](#)

Jan 2024 - May 2024

Metro Vancouver, Vancouver - *Unity VR Developer*

This project won [Gold and Silver Horizon Interactive Awards](#) and was showcased at the **Pacific National Exhibition (PNE)**. Key contributions:

- Developed the **VR** experience in Unity's **XR Interaction Toolkit** and **C#**.
- Integrated the **ArcGIS SDK** into the experience as the client requested.
- Collaborated with a cross-disciplinary team of designers and artists in an agile environment.
- Documented the technical implementation for future development handover.
- Transformed the client's requirements and feedback into actionable goals.

Sep 2023 - May 2024

Centre for Digital Media, Vancouver - *Instructional Assistant*

- Reviewed students' assignments and provided feedback whenever needed.
- Assisted faculty in the preparation and delivery of course materials.

Projects

Tank Lore - [Steam](#) | [GitHub](#) | [Project Link](#) | [Video Link](#)

Unity | C#

A rogue-like game inspired by Battle City and Metal Slug, built on a 1,000+ script, 60K-line codebase. My key responsibilities include:

- Implemented most of the game's systems, including random level generation, the map creation tool, and local multiplayer (all listed in my [portfolio](#)).
- Architected the codebase for scale, keeping systems modular and maintainable as the project grew.
- Designed test plans and ran playtests, using the feedback to drive bug fixes, gameplay troubleshooting, and optimization.

Swordsluke - [GitHub](#) | [Project Link](#)

Unreal Engine | C++ | Multiplayer Subsystems | Steam Subsystems

A multiplayer souls-like game, fully written in C++. Key tasks and features include:

- Implemented fully functional multiplayer gameplay and session creation using **Steam Subsystems**.
- Developed an advanced combat system with lock-on, combo attack queues, attack indicators, a parry system, stamina and health systems, and dodge rolls.
- Created a custom multiplayer sessions UE package and a set of reusable combo montage notifies.

Behavior Tree System Framework (Unity Tool) - [GitHub](#) | [Project Link](#)

Unity | C# | UI Toolkit | XAML

Developed a custom AI behavior framework in Unity using Graph View, Editor Window, Custom Editor, and UI Toolkit libraries. The tool is performance-optimized and tested in production on Tank Lore, where it streamlined the implementation of complex AI behaviors.

Meteoric Wars - [GitHub](#) | [Project Link](#) | [Video Link](#)

Unity | C#

A space shooter rogue-like inspired by Sonic Wings. Managed end-to-end development, including core features such as procedural ship generation, a dynamic combat system, UI design, and art assets.

PlayCanvas/GitHub Sync Tool - [GitHub](#) | [Video Link](#)

C# | .NET | WPF | MVVM | REST APIs

A C# .NET WPF application that automates syncing PlayCanvas projects with GitHub repositories, built to keep a record of my work on GitHub without manual effort.

- Developed flexible sync options for downloading and pushing PlayCanvas builds, assets, or both to GitHub.
- Designed a user-friendly configuration interface following the **MVVM** pattern for maintainable, scalable UI.
- Implemented detailed logging with real-time updates and automatic log file generation.
- Utilized RESTful APIs to interact with the PlayCanvas API and retrieve required data.

Unity Tools

A list of Unity tools I developed to make life using Unity easier:

- **Notion Linker**: links Unity's scriptable objects to databases on Notion.
- **Method Button** [\[Link\]](#): an attribute that displays methods in code as buttons in the inspector.
- **Regions Screenshots** [\[Link\]](#): captures multiple screenshot regions from a single camera.
- **Enhanced Hierarchy** [\[Link\]](#): organizes GameObjects in the Hierarchy window using colors and icons.

Skills

Programming Languages and Frameworks: C# | C++ | .NET | WPF | XAML | JavaScript | HTML | CSS | Python | SQL

Game Engines: Unity | Unreal Engine | PlayCanvas

Others: Git | GitHub | Visual Studio | Rider | Jira | Notion | Trello | Photoshop | Blender | Audition | Excel | WebStorm

Soft Skills: Problem-solving | Teamwork | Communication | Adaptability | Attention to Detail | Creativity | Constructive Feedback

Education

2022 - 2024/2025

Master of Digital Media - Simon Fraser University, Canada

2017 - 2021

Bachelor of Information Technology - University of Aden, Yemen