

Personal profile

Junior Gameplay / Systems Programmer with a BSc in Game Software Engineering, specialising in C++ gameplay systems, engine-level programming, and Unreal Engine 5 development. Experienced in building modular game systems, custom engines, and replicated gameplay features. Strong foundation in OOP, state machines, and performance aware design, with a focus on clean, maintainable code and playable features. Fluent in English, Italian, and Arabic.

TECHNICAL SKILLS

- **Languages:** C++, C#, Python
- **Frameworks/Engines:** Unity, Unreal Engine 5, SDL2 (engine development)
- **Tools:** Git, GitHub, Debugging, version control, Agile / iterative development
- **Gameplay & Systems:** OOP, State Machines, Multiplayer Replication fundamentals, Data driven design, Input systems, Collision & physics handling

EDUCATION

BSc (Hons) Game Software Engineering — Bournemouth University (2021–2025)

Key modules: AI Systems, Game Programming, Software Design, Physics Simulations

PROJECT EXPERIENCE

Rum Runner's Revenge — C# | Unity | [Playable Demo & AI scripts](#) Jan - May 2024

- Contributed to a 2D pirate-themed action RPG developed by a team of 11 over 3 months.
- Designed and implemented enemy AI using a custom finite state machine, managing patrol, chase, and attack behaviours.
- Developed physics-based gameplay systems, including a collapsing bridge and in-game mini-map.
- **Achievement:** The project was nominated for the TIGER Game Award at Bournemouth University.

2D Arcade Game Engine — C++ | SDL2 | [Project Link](#)

Sept - Ongoing 2025

- Built a modular 2D game engine with systems for rendering, collision detection, and input handling.
- Designed reusable architecture using OOP principles for maintainability.
- Implemented sprite batching and modular systems to improve performance and scalability.
- **Achievement:** Developed strong C++ fundamentals and engine level system design skills.

Aura Multiplayer RPG — C++ | Unreal Engine 5 | [Project Link](#)

Jun - Dec 2024

- Developed a replicated multiplayer gameplay system supporting synced movement and interactions.
- Built UI elements and input mapping, integrating C++ and Blueprint logic.
- Implemented state driven behaviour for consistent, predictable gameplay flow.
- **Achievement:** Built and tested replicated multiplayer gameplay systems.