Miguel Fiuza Gomes

Graduate Game Developer (Systems / Gameplay Programming)

Lisbon, Portugal
miguel.fiuza.gomes@gmail.com • https://miguelfiuzagomes.github.io/portfolio/
github.com/MiguelFiuzaGomes • linkedin.com/in/miguel-fiuza-gomes

---

Profile

Graduate game developer specialising in procedural generation, real-time systems, and engine-level programming. Experienced in designing modular generation pipelines, implementing networked gameplay architectures, and building rendering systems using **Unity**, **Unreal Engine**, and **DirectX 11**. Strong focus on performance-aware design, scalability, and robust system architecture.

Technical Skills

- **Languages:** C#, C++, HLSL
- **Engines & Frameworks:** Unity, Unreal Engine, DirectX 11
- **Core Areas:** Procedural Generation, Multiplayer Networking, Real-Time Rendering, Physics Systems, Performance Optimisation, Systems Architecture
- **Tools:** GitHub, Visual Studio

Selected Projects

Procedural Terrain Generation System (MSc Project) — Unity, C#

- Architected a modular procedural pipeline separating **terrain generation**, **biome evaluation**, and feature application, enabling **scalable** and **extensible** world generation
- Implemented rule-based biome classification using **domain-warped** noise fields (temperature, humidity, elevation), enabling controlled biome distribution across large-scale procedural environments
- Developed infinite terrain streaming with **chunk-based generation**, including **normalisation** and **seam handling** across chunk boundaries
- Integrated **shader-based biome blending** to transition from discrete biome assignment to continuous visual output
- Enabled generation of **large-scale**, diverse terrain with **controlled biome distribution** and **seamless chunk** transitions at runtime

Multiplayer FPS Netcode Prototype — Unity, Netcode for GameObjects

- Designed a **semi-authoritative** networking architecture balancing client-side responsiveness with authoritative server validation to maintain consistency under network latency
- Implemented **client-side** movement and shooting with **server reconciliation** for **divergence correction**
- **Synchronised player state** (position, rotation, actions) across networked clients in real time
- Analysed **latency**, **prediction errors**, and **desynchronisation** during testing and iteration

DirectX 11 Real-Time Rendering Pipeline — C++, HLSL

- Implemented a real-time **rendering pipeline** from scratch, including shader management and lighting systems
- Developed **HLSL shaders** and managed GPU/CPU interaction, including buffer updates and rendering flow
- Worked directly with **graphics pipeline** stages, gaining low-level understanding of **rendering** flow and **GPU** constraints

Snowboarding Physics Prototype (VeilRyders) — Unreal Engine

- Implemented **slope-aware** movement using **surface normals** and **velocity vectors** to achieve **physically coherent** traversal behaviour
- Tuned **physics** parameters to balance realism with responsive and controllable gameplay
- Iterated on movement systems to improve **player feedback** and consistency
- **Collaborated** with designers and engineers to iterate on movement feel and gameplay responsiveness

---

Education

MSc in Computer Games Technology

Abertay University — 2024-2025

BSc in Games and Apps Development

IADE — 2019-2023

Languages

English - C2 awarded by the British Council

Portuguese - Native