Luca De Meyer

GAME DEVELOPER

Luca.de.meyer15@hotmail.com | +32 471 78 70 68
Portfolio: http://lucademeyer.com | LinkedIn: https://www.linkedin.com/in/luca-de-meyer-05b360295

About

Final-year Game Development student at Howest DAE. My portfolio spans the entire tech stack, from building a custom Vulkan rendering engine to programming decoupled UE5 gameplay systems. Because I have experience with both low-level engine architecture and high-level AI logic, I can quickly adapt to different proprietary codebases. I am seeking a Spring 2027 programming internship where I can act as a versatile generalist, jumping in to debug or build features wherever the engineering team needs support.

TECHNICAL SKILLS

Languages:

- Proficient: C++, C#, C, Python, HLSL, GLSL
- Familiar with: 6502 Assembly

Engines & APIs: Unity, Unreal Engine 5, OpenGL, Vulkan, Custom engine development, DirectX11
Software & Tools: Git, Visual Studio, Maya, Photoshop, Substance painter, Perforce, Reaper
Specialties: Gameplay Programming, Graphics Programming, VR Development, Systems Architecture

---

TECHNICAL & GAME PROJECTS

Graphics Programming Thesis: Hybrid Ray Tracing | Solo Researcher/Programmer | September 2025 – August 2026

- Engineered a custom hybrid Vulkan renderer in C++ and GLSL, integrating real-time ray-traced shadows, reflections, and ambient occlusion alongside a deferred rasterization baseline.
- Architected a data-driven modular render graph to automate complex Vulkan synchronization and VkImageMemoryBarrier dependency management across compute and raster pipelines.
- Implemented a 5-level spatiotemporal denoiser (SVGF) and profiled inline (rayQueryEXT) versus pipeline ray tracing overhead, optimizing the full RT execution down to 2.24ms per frame.

Custom 2D Game Engine | Engine programmer | February 2026 – June 2026

- Expanded a foundational C++ and SDL2 framework (Minigin) into a robust 2D game engine, implementing a fixed and variable timestep game loop, thread-safe logging, and a resource manager.
- Architected a flat component-based GameObject system and implemented core design patterns (Observer, Command, Service Locator) to strictly decouple gameplay logic from UI and input.
- Optimized memory management using custom allocators and Object Pooling for projectiles and

particles to maintain strict performance during high-density gameplay.
- Validated the engine by shipping a multi-mode recreation of *Tron (1982)*, complete with state-machine AI and abstracted input logic for full controller support.

Educational VR Simulator (SMBG Client Project) | VR Developer | September 2024 – December 2024

- Executed a mid-production engine migration from UE4 to UE5, updating legacy C++ APIs, integrating Chaos Physics, and porting controls to the Enhanced Input System.
- Programmed complex, state-based VR interactables and a centralized objective management system to drive a repeatable gameplay loop on Meta Quest 3.
- Collaborated in a cross-disciplinary team of 5 to deliver a 1-to-1 high-fidelity simulation for a real-world client, strictly optimizing logic to maintain 90 FPS on standalone hardware.

Go To Hell (Co-op Twin-stick shooter) | Gameplay programmer | September 2023 – December 2023

- Programmed a segmented, multi-phase boss fight and distinct enemy AI archetypes (ranged and melee) using C++ in UE5.
- Architected a decoupled wave management system, enabling game designers to tune spawn frequencies and difficulty curves independently from core AI logic.
- Collaborated within a cross-functional team of 4 throughout a full production cycle, aligning technical implementations with design and art bibles.

EDUCATION

Howest Digital Arts & Entertainment (DAE)
Bachelor's degree in Game Development | September 2021 - present