

NOYA CAI

qimiaocai011110@gmail.com ◇ +1-5853175388 ◇ noyacai1110.github.io/Portfolio-Website
Rochester, NY ◇ linkedin.com/in/noya-cai-124489292 ◇ github.com/NoyaCai1110

OBJECTIVE

Gameplay/Graphics programmer and Technical artist. Available now. Open to relocate.

EDUCATION

Rochester Institute of Technology (RIT)

Aug/2023 - May/2025

Master of Science in Game Design and Development

Course Taken: [Computer Animation](#), [Game Graphics Programming](#), [Global Illumination](#)

University of Science and Technology of China (USTC)

Sep/2019 - June/2023

Bachelor of Engineering in Computer Science

Course Taken: [Computational Methods](#), [Equations of Mathematical Physics](#), [Data Structure and Algorithm](#)

SKILLS

Programming	Proficient in C, C++, C#, GLSL, HLSL, Python, Swift, Java, HTML
Library	OpenCV, OpenGL, OpenXR, DirectX11, DirectX12, CUDA, ARKit, RealityKit
Software	Unity, Unreal Engine, Visual Studio Code, Visual Studio, Xcode, Trello, Figma

WORK EXPERIENCE

Ylem Studio, Gameplay Engineer

May/2025- Present

Keywords: [VR](#), [PCVR](#), [Unreal Engine 5](#), [C++](#), [Gameplay Systems](#), [Physics](#), [Performance Optimization](#)

- Developed gameplay systems and interactive mechanics for a VR puzzle game using the Oculus branch of Unreal Engine 5, an open-source VR-optimized UE5 build.
- Implemented VR interactions, physics-based manipulation, and puzzle logic using Blueprints and C++.
- Designed modular gameplay systems that support multiple player-driven solutions and emergent gameplay.
- Optimized gameplay performance, memory usage, and interaction responsiveness for both standalone VR and PCVR platforms.

Magic Spell Studio, Augmented Reality Software Engineer

Jan/2024-Apr/2025

Keywords: [VR/AR](#), [Unity](#), [C#](#), [SwiftUI](#), [ARKit](#), [RealityKit](#), [ios](#), [Figma](#), [Trello](#)

- Developed an AR windowed application using Unity for Apple Vision Pro to be used in medical fields.
- Worked with designers to implement complex UI/UX system to meet client's needs.
- Worked with data engineers to migrate data from FHIR server to an AR application.
- Using SwiftUI, ARKit, Compositor Services, and RealityKit to develop an AR immersive application for Apple Vision Pro to be used in medical research.
- Separated the main thread of the program into multiple threads and improved the overall speed by 30%.

PROJECTS

Duolatera (Capstone VR online co-op puzzle game)

Aug/2024-May/2025

as [Technical Artist](#), and [Graphics/Gameplay Programmer](#), using [UE5](#), [C++](#), [HLSL](#) and [Unreal Insight](#)

- Designed and implemented a cel-shading pipeline for forward rendering, which is otherwise unachievable through regular methods commonly used in deferred rendering
- Using both materials and Niagara systems, created procedurally generated VFX for portals, lasers, shoot-able items, interact-able items as well as several other in-game props.
- Implemented in-game dialogue system, voice chat system, load and save game system, and several other gameplay puzzle mechanics.
- Improved the overall game performance by analyzing data from Unreal Insight and optimizing gameplay and shader code complexity.

DirectX11 Particle System

Apr/2025-May/2025

[as Graphics Programmer, using C++, DirectX11, HLSL](#)

- Developed a real-time particle system with flexible, extensible emitter architecture supporting multiple simultaneous particle effects using DirectX 11 and C++.
- Implemented both sprite-based and mesh-based particle emitters, customizable materials, and beam rendering to enable diverse visual effects within a single system.
- Built CPU/GPU hybrid simulation where per-particle behavior is computed efficiently on the GPU using shaders, enhancing performance and scalability for large particle counts.
- Used shader-based billboard and dynamic buffer techniques to render particles as camera-facing quads without traditional vertex buffers, demonstrating low-level graphics pipeline mastery.

DirectX12 Real-Time Path Tracer

Jan/2025-Apr/2025

[as Graphics Programmer, using C++, DirectX12, HLSL](#)

- Upgraded an open-source DirectX11 renderer to DirectX12 and implemented real-time GPU path tracing.
- Redesigned the rendering pipeline and integrated bindless texture access for greater flexibility and scalability.
- Applied GPU-driven techniques to optimize performance for real-time rendering of complex scenes.

Ocean Simulation Shader

Feb/2024-Apr/2024

[as Graphics Programmer, using C++, GLSL, OpenGL](#)

- Developed a real-time ocean simulation shader with ambient waves, interactive ripples, procedural sand, and GPU-based lighting.
- Implemented Gerstner wave function for ambient waves and circle wave function for interactive ripples, creating realistic surface movement and buoyancy effects.
- Generated ocean floor sand using Perlin noise and integrated a GPU raytraced lighting system with multiple reflection and refraction for realistic rendering.