

RYAN RAMOS

Riverside, CA | ryannr9375@gmail.com | (718) 427-0741 | [Portfolio](#) | [LinkedIn](#)

SUMMARY

Fourth-year Computer Science student and gameplay programmer with experience developing AI, gameplay systems, and designer-facing tools at an Xbox Game Studios developer. Passionate about creating player-focused experiences through thoughtful engineering, rapid iteration, and close collaboration with designers and fellow developers. Driven to continuously improve my craft and learn new things, learn from feedback, and build reliable systems that help teams deliver memorable experiences to players.

PROFESSIONAL EXPERIENCE

Obsidian Entertainment, Inc. (XBOX Game Studios)

Irvine, CA | Jun 2026 – Aug 2026

As an AI Engineering Intern at Obsidian Entertainment, I contributed to gameplay AI systems and module-level system refactors during critical stages of production. Over 12 weeks, I collaborated with engineers, designers, and other disciplines to improve runtime behavior, debugging workflows, and content iteration by delivering reusable, maintainable systems with project-agnostic debugging tools and AI behaviors.

AI Engineer - Intern

Jun 2026 – Aug 2026

- Contributed to **200+ code reviews**, providing and incorporating feedback on code quality, correctness, and maintainability.
- Aided in debugging and resolving errors and warnings, reducing 24-hour diagnostic log volume by **99.98%** from **~500,000 to 100** entries and preserving actionable telemetry for analysis.
- Developed a **runtime audio-stimulus visualization tool** supporting 25+ unique sound types by extending Unreal Engine's Gameplay Debugger framework, enabling area designers and AI engineers to validate audio-driven gameplay behavior in real time.
- Refactored AI tasks from per-frame ticking to **delegate-driven callbacks**, creating project-agnostic, reusable behaviors organized across modules.
- Utilized **non-unity batch builds** and **local cooks** to identify errors before submitting changes to the development stream.
- Converted core gameplay Blueprint behavior classes into native C++ AI tasks.
- Retired **three tightly coupled legacy systems** while preserving project functionality.
- Implemented console commands that toggle visibility of certain HUD elements based upon specific gameplay conditions.

Gameheads Oakland

Oakland, CA / Remote | Jun 2020 – Present

Over five summers at Gameheads, I developed six game projects, including one released on Steam. I worked as a Gameplay Engineer on multidisciplinary teams of five to seven. Through daily SCRUM collaboration, I worked closely with designers and artists to build custom tools, resolve technical challenges, and helped streamline multidisciplinary workflows in version-control.

Systems & Tools Engineer | Pliant

Jun 2025 – Jun 2026

- Engineered an **event-driven objective system** supporting 10 objective types and 30+ gameplay events to decouple gameplay events from quest progression by using the strategy pattern.
- Developed custom in-engine, **designer-facing editor tools** for authoring player-facing game and quest content.
- Integrated UI components with gameplay events, player key-binds, and objective data, enabling real-time **multi-input support** updates on player progress and game events.

Gameplay Programmer

Jun 2020 – Present

- Spearheaded the development of **multiplayer gameplay systems** for tactical shooter *PROJECT: Black Cat*, including replicated game state, RPC-driven abilities, and vector-based weapon hit calculations.
- Reduced frame time by 85% (33ms to 5ms)** during enemy-heavy encounters of 50+ enemies by caching shared A* pathfinding results and pooling spawned enemies in Unity horde shooter, *Claim to Fate*.
- Selected as Lead Programmer; contributed to **50+ code reviews** and conducted weekly 1-on-1 mentoring sessions, supporting 20+ programmers over one year.

Personal Project | Unreal 5 - Gameplay Sandbox

Present

- Developed **custom C++** AI perceptions, controllers and task classes for responsive enemy behavior with Behavior Trees.
- Engineered player detection logic in C++ by utilizing squared distance, field-of-view, and line of sight checks to transition AI between states.

ACCOMPLISHMENTS

- Shipped [Pliant](#)** on Steam as part of a six-person multidisciplinary team.
- 2023 Unity Student Showcase - Featured Project (*Pliant*)

SKILLS & TOOLS

Languages: C++, C#

Engines: Unreal Engine 5, Unity Game Engine

Tools: Perforce, Helix Swarm, Unreal Game Sync, Git, JIRA, Confluence

EDUCATION

California State University, San Bernardino

San Bernardino, CA

BS, Computer Science (May 2027) | Honors Program, Dean's List (2025-2026)