

Arya Hosseini

Montreal, QC, Canada

(514) 662-1089

[Portfolio](#) , [GitHub](#), [Linkedin](#)

hosseini.arya.8910@gmail.com

SUMMARY

Game Programmer proficient in working with Unreal Engine, Unity and proprietary engines. Experienced in developing engine features, gameplay systems and tools. Driven by writing performant, maintainable and high quality programs.

SKILLS

Engines & Tools: Unreal Engine, Unity, Git, ImGui, SFML, LLDB, GDB, Tracy Profiler, CMake, Maya

Programming Languages: C++, C#, Lua

EXPERIENCE

Unity Gameplay Programmer Intern

Baobab Games | Montreal, Canada | May 2024 - Aug 2024

- Designed an item storage system integrated with the player inventory and UI, optimized with memory pooling and custom physics behaviours, delivering a stable, performant feature that shipped to playtests.
- Prototyped and integrated 18 new building parts into the base building system, each with designer defined constraints and unique gameplay interactions, expanding available building options for players by over 50%.
- Debugged and profiled gameplay systems alongside the programming team to hit performance targets, contributing to code reviews and documentation, resulting in cleaner, more maintainable code across the codebase.
- Collaborated with artists, designers, and QA in an Agile environment to diagnose and resolve gameplay issues, keeping cross discipline communication clear throughout the development cycle.

PROJECTS

ECS Game Engine (C++ 20)

[View Project](#)

- Built a data oriented 2D game engine from scratch with a custom ECS core, organizing entities into memory efficient chunks with cache friendly layouts and SIMD friendly data structures, making large scale entity iteration significantly faster.
- Implemented a custom memory allocator with cache line aligned heap allocations and type erased block storage, eliminating memory fragmentation and heap overhead in hot loops.
- Designed a centralized component registry to coordinate cross thread entity operations safely, enabling multithreaded reads and structural mutations without data races or synchronization bugs.
- Created a data driven prefab system with JSON serialization, allowing entities to be defined, modified, and instantiated entirely through the editor without touching code.
- Built a query system with required and excluded component filtering across archetypes, letting systems iterate only over relevant entities with no runtime overhead.
- Built a two phase physics pipeline: a spatial grid broad phase to eliminate irrelevant collision candidates early, followed by AABB narrow phase with penetration resolution and a layer/mask bitmask system for precise collision filtering.
- Integrated Tracy Profiler with zone scoped markers and LLDB for low level debugging, enabling frame level CPU bottleneck identification and runtime crash diagnostics through a live log window with stack trace tracking.
- Implemented a batch rendering pipeline over OpenGL, grouping sprite draw calls by texture using manual vertex arrays, reducing GPU state changes and draw call overhead as entity counts grow.
- Developed a Dear ImGui editor with an entity inspector, rendering viewport, and live log window, enabling full in editor scene authoring and real time debugging without interrupting the engine loop.

Action Roguelike (Unity)

[View Project](#)

- Designed an OOP character controller using SOLID principles and established design patterns, producing modular gameplay systems that are straightforward to debug, extend, and profile.
- Built a runtime procedural world generation system using zone based architecture and binary space partitioning with proximity based culling, ensuring seamless exploration at any world size.
- Replaced Unity's built in NavMesh with a custom Flow Field pathfinding system tailored for dynamic procedural environments, enabling reliable AI navigation across layouts that change at runtime.
- Implemented an enemy AI framework driven by ScriptableObject defined behaviours, producing reusable, configurable AI states that designers can extend without programmer involvement.

EDUCATION

DEC – Computer Science, Video Game Programming

LaSalle College | Montreal, Canada | 2022 - 2024