ASHER MARSEE

Software Engineer · Game Developer

natemarsee@icloud.com · (425) 443-2158 · LinkedIn · Portfolio · GitHub (nathanmarsee)

Professional Summary

Recent computer science graduate and self-driven software engineer with hands-on experience across the full development lifecycle: requirements gathering, system design, full-stack implementation, automated testing, and iterative delivery. Demonstrated ability to rapidly acquire unfamiliar technologies and frameworks and apply them to real-world problems under deadline pressure. Experienced in Unity-based game development, full-stack web development (.NET/Blazor, SQL, Azure), and agile team leadership. Equally comfortable working independently on complex technical problems or working within a cross-functional team toward a shared product vision. Strong background in UI/UX design, accessibility, and user-centered iteration. Motivated by technically challenging problems, emerging technologies, and building software that measurably improves the experience of its users.

Education

B.S. Computer Science | *Eastern Washington University* April 2022 – December 2025
Cheney, WA

Four-year degree completed December 2025. Relevant coursework included Data Structures and Algorithms, Compilers, Operating Systems, and .NET Web Application Development.

Leadership: Served as treasurer and de facto organizer of the EWU Game Club (Spokane Branch) for approximately two years, managing scheduling, logistics, and administrative paperwork for weekly meetings.

Honors College | *Western Washington University* August 2020 – December 2021
Bellingham, WA

Completed foundational coursework prior to transferring to EWU. Relevant courses included Computer Systems (C and Unix), Linear Data Structures, Functional Programming, and Physics with Calculus I–III.

Work Experience

Software Engineering Intern | *UW Medicine – Neitz Lab* June 2019 – August 2019
Seattle, WA

Context & Problem

Joined a vision science research lab studying the effects of high-contrast visual stimuli on myopia (near-sightedness). The research focused on the choroid, a vascular layer of the eye whose swelling is clinically linked to the progression of near-sightedness. My initial role was to manually segment and highlight the choroid region in eye scan imagery using Adobe Photoshop, producing clearly labeled training data to develop a machine learning model (AI) capable of automating that annotation work.

Identifying an Opportunity

Participants in the vision experiments were required to play a single virtual reality table tennis game for the duration of each session, up to an hour. Engagement was declining, and the existing game offered limited ability to tune the visual contrast parameters that the experiment required. During a team lunch, I mentioned my experience developing games in Unity; the researchers immediately recognized an opportunity and invited me to design an entirely new VR game tailored to the experimental requirements.

Technical Execution

Working independently and largely self-directed, I designed and shipped a complete original VR game called Jet Slalom VR within a two-week window. This was my first complete, original game development

project; prior Unity experience was limited to small personal explorations. I maintained velocity by keeping scope deliberately constrained: a single-axis infinite runner with aim-and-fire shooting mechanics and a high-score objective, eliminating feature risk while preserving replayability and engagement.

The most significant design challenge was motion sickness: early playtesters reported nausea from the rotation of the spacecraft. Rather than restricting the flight mechanics, I implemented an in-cockpit secondary display showing a fixed third-person perspective of the ship. This gave nausea-prone players a stable visual anchor, effectively a "TV mode," while preserving the full experience for other players. The solution also doubled as a tactical tool for advanced play, demonstrating that accessibility features and depth of gameplay need not be in tension.

Research-Aligned Visual Design

The visual design was engineered to serve the experiment's scientific requirements. The background and obstacles used maximum-saturation red and blue on a high-contrast grid, with bright yellow projectiles as the primary interactive elements. This ensured that the points of highest contrast were always at the center of the player's attention rather than the periphery, a deliberate inversion of the previous table tennis game's design. Because high-contrast stimuli were more consistently present throughout play, researchers could draw stronger conclusions from the experimental data.

Outcome

Jet Slalom VR was adopted for daily use in the lab's experimental sessions. Player engagement was strong enough that participants spontaneously organized their own informal leaderboard, an unsolicited signal of genuine replayability. Researchers reported improved confidence in data quality as a result of the more consistent high-contrast visual environment the game provided.

Technologies: Unity (C#), Meta (Oculus) Rift & Quest VR hardware, Adobe Photoshop

Projects

HeroScape Digital Scenario Editor | *EWU Senior Project (Lead)* January 2025 – June 2025

Stack: Unity (C#) · Project Portfolio Page

Overview

Led a team developing a real-time 3D level editor for HeroScape, a hex-tile tabletop game with an active community of custom scenario designers. Existing community tools were universally outdated: most were 2D-only (requiring manual layer switching to work in three dimensions), non-real-time (requiring an explicit render step to preview placements), unintuitive, and visually dated. The goal was to deliver a modern, real-time 3D editor with interaction conventions familiar to users of tools like Blender or the Unity editor itself.

Core Technical Challenge: Hexagonal Terrain Placement

The central engineering problem was implementing correct, robust placement of hexagonal terrain pieces on a 3D hex grid, including automatic snapping, stacking detection, and piece rotation. Hexagonal coordinate math, while non-standard, was well-documented and implementable with straightforward research. The harder problem was reliably tracking occupancy state across the grid: which hex columns were filled at which vertical levels, and ensuring that state remained consistent across piece placement, removal, and rotation operations.

Team Leadership Under Adverse Conditions

The project was structured as a four-person senior project. Early in development it became clear that the team was not taking initiative: sprint goals were going unaddressed with no communication, and submitted work was rushed, incomplete, or did not address the stated requirements. One team member ultimately ceased all participation a quarter of the way through the project, reducing the effective team to two, and in practice, given the volume of feature implementation I carried, closer to 1.5.

Rather than allowing the project to stall, I transitioned from a collaborative leadership model to an active assignment model: explicitly delegating specific, scoped tasks to the remaining contributor, conducting structured code reviews, and providing detailed written feedback when submissions did not meet requirements. I simultaneously absorbed the majority of feature development myself while continuing to

fulfill the design director and project manager roles. The project was delivered successfully as a complete, functional senior project by the June 2025 deadline.

Requirements & Design Process

Prior to development, I conducted user research with the active HeroScape scenario design community, interviewing designers about their existing workflows and pain points. This confirmed that 3D real-time visualization and modern, intuitive controls were the highest-priority gaps in the tool landscape. I produced comprehensive design documentation covering user stories, technical specifications, and UX rationale, establishing a shared reference that kept the project aligned across the team throughout its six-month development arc.

Outcome

The editor was completed to a solid, functional state by the end of the senior project. Public release is planned following a period of further polish and development. The project demonstrated the viability of a modern, accessible 3D editing experience for the HeroScape community.

Ball World & Astroll | *EWU Class Projects (Lead)* April 2025 – June 2025

Stack: Unity (C#) · Project Portfolio Page

Overview

Pitched and delivered two separate physics-based 3D platformer games across two sequential four-week development cycles, each on a different three-person team. Both games used a tilt-based control scheme I had originally developed as an independent project in 2020, inspired by the Super Monkey Ball franchise, a system in which the player tilts the stage itself to influence the ball's momentum rather than directly controlling the character, producing distinctive physics-driven gameplay with significant depth.

Rather than building the control system from scratch under a short deadline, I brought proven, well-understood code as a foundation and directed both teams' design efforts around it. This was a deliberate decision to manage scope risk and maximize the time available for content and polish.

Ball World – Infinite Runner (Sole Developer for Core Mode)

Conceived and implemented the infinite runner game mode independently. The central design challenge was player incentive: in a pure survival mode, optimal play is slow and cautious, which produces unengaging sessions. Drawing on the design language of foundational arcade games, I implemented a dual-incentive system: coins placed along high-risk, high-speed corridors, and a time bonus awarded for fast segment completion. This created meaningful tension between the safety of conservative play and the score ceiling of aggressive, high-speed routes, producing a higher skill ceiling and more replayable sessions without additional content.

Astroll – Level Design Lead

On the Astroll project, I served as level design lead and personally authored approximately one-third of the game's levels. I also acted as the primary playtester, iterating on level design to ensure fairness and engagement across the full difficulty range.

I was also responsible for implementing a freely-controllable camera system when Astroll's design requirements called for one that the original control system did not include. This was developed over a few days and integrated cleanly with the existing physics framework.

Both projects were completed within their respective four-week delivery windows.

Wordle Recreation | *EWU Class Project (Full-Stack)* May 2024 – June 2024

Stack: .NET (C#), Azure (SQL, hosting, authentication), Vue.js (TypeScript), GitHub Actions

Overview

Collaborated with a partner to build a full-stack web application recreating the New York Times Wordle game. Assumed primary responsibility for the backend: designed and implemented a SQL database schema to store the daily word and per-user game statistics, built a server-side REST API in C# using the .NET framework to serve data to the client, and configured hosting and user authentication through Microsoft Azure.

Established a basic CI/CD pipeline using GitHub Actions with automated .NET unit tests running on each build, ensuring regressions were caught at integration time. Assisted my partner who had limited prior experience with frontend frameworks in connecting the Vue.js frontend to the backend API, including code review and pair debugging sessions. The most technically friction-heavy aspect of the project was navigating Azure's configuration interface, which required persistence and careful documentation work to get the hosting and auth pipeline functioning correctly.

Orbellion Web | *Independent Project (Lead)* Ongoing

Stack: Blazor/Razor Pages (C#), JavaScript, HTML/CSS, SQL, SignalR (in progress) · GitHub

Problem Statement

Perfect Draw! is a tabletop RPG with a card-based combat system. Playing virtually requires a shared digital interface for managing players' hands, decks, and boards, as well as the ability for the game master to create and edit cards on the fly for live balance adjustments. Existing virtual tabletop platforms lacked accessible, real-time card creation and editing capabilities, so I initiated development of a purpose-built web application to fill that gap.

Current State

The application is in late-stage development. Completed features include a full card creation and editing interface (Blazor/Razor), user authentication with persistent card storage (Blazor/SQL), and an interactive board and card display interface (JavaScript). Multiplayer interface designs are finalized; the remaining primary implementation work is real-time multiplayer synchronization via SignalR.

Team & Process

Leading a four-person volunteer team with varied schedules and skill levels. The informal nature of the project and the absence of hard external deadlines have occasionally made sustained momentum a challenge; I have managed this by maintaining team engagement through regular communication, keeping the remaining scope clearly articulated, and being prepared to absorb additional implementation work during periods when my primary co-developer's professional schedule constrains their availability.

This project has also served as a deliberate opportunity to integrate AI-assisted development tooling into my workflow, using Claude Code and GitHub Copilot for boilerplate generation and rapid API exploration with the goal of building fluency with these tools as a standard part of professional development practice.

Technical Skills

Languages

C# (primary), JavaScript / TypeScript, SQL, Python, Java, C/C++; strong record of acquiring new languages and frameworks rapidly as project requirements demand.

Frameworks & Platforms

.NET Core, Blazor / Razor Pages, Vue.js, React, HTML5 & CSS3, Unity (game engine), SignalR.

Infrastructure & Tooling

Microsoft Azure (hosting, authentication, SQL), GitHub / Git (version control), GitHub Actions (CI/CD pipelines), Visual Studio, IntelliJ, Eclipse; Windows and Linux operating systems.

AI-Assisted Development

Active user of Claude Code and GitHub Copilot for boilerplate acceleration, API exploration, and rapid onboarding to unfamiliar libraries. Treats AI tooling as a force-multiplier for productivity rather than a replacement for engineering judgment.

Design & Creative Tools

Adobe Photoshop (10+ years, advanced proficiency), Adobe Illustrator, Affinity suite; applied to digital art, texture and asset creation, and high-fidelity custom card design for print. UI/UX design principles, accessibility-first design, rapid prototyping and playtesting methodology.

Core Competencies

Full-stack web development · Agile/Scrum team leadership · System and API design · Automated testing · CI/CD pipeline setup · VR application development · Game design and mechanics prototyping · User research and requirements documentation · Technical communication across skill levels.

Domain Interests

Game design and game engine development · Virtual reality technology · UI/UX and interaction design · Graphics programming (shaders, lighting, rendering pipelines) · Vision science and healthcare technology · Machine learning applications in medical imaging.