Seth Anderson

218-340-7768
seth.anderson32@gmail.com
about.s10n.onl

Objective

To work on games and improve as a software engineer. Willing to relocate.

Langauges

C++ (16yrs), C# (14yrs), Javascript (10yrs), SQL (8yrs), C (5yrs), Java (1yr)

Games-Focused Highlights

- Most recently worked on an indie MMO in Unity (see entry under Games Experience)
- Extended personal experience with C++: many game-related projects, some GUI applications (usually using the Qt framework), have used: obviously stdlib (and some stdlib alternatives), SDL2/SDL3, SDL_image, SDL_ttf, Tracy Profiler, OGRE3D 1.x, Squirrel, OpenAL, enet, Sqlite, and a bunch I've probably forgotten
    - One such project early in my career: a C++ game engine using OGRE3D, enet, OpenAL, Squirrel, a built-in all-in-one editor made in Qt with a scene editor, script editor, and model viewer, with component-based game objects and a pub-sub event system
    - An ECS-based engine experiment in C-style C++ focused on memory contiguity and ease of utilizing SIMD intrinsics (more on that in the Significant Personal Projects section)
    - Many other small and/or inconsequential projects over the years
- Class projects using the Torque3D engine and its scripting language: TorqueScript. Worked with multiple artists to bring their work into the projects (2D and 3D assets)

Games Experience

Game Programmer (Contract) – Stormhaven Studios – May 2020 - Present

Stormhaven Studios is an indie development studio working on our debut title Embers Adrift which was released in 2022. Embers Adrift is an MMORPG made in Unity.

The only other full-time programmer aside from our CTO working on core game systems

- Implemented and maintained our crafting system: a relatively complicated solution allowing flexible use of materials per recipe and allowing those materials to ultimately affect the end stats of an item that may be several crafting steps removed. I worked extensively with one of our designers to add features and capabilities throughout development
- Responsible for our social service—which handles chat, groups, raids, LFG/LFM, friends, guilds, block lists, invites, mail, and cross-zone player status—as well as all the associated UIs. This service was implemented in C# using modern .NET, hosted on a Linux machine within a Docker container and utilizing the TPL to perform most actions asynchronously
- Monitored logging and player reports to find and address problems as they arose, including multiple investigations to determine whether certain events were caused by logic errors or player behavior
- Designed and built our quest system and associated UI, integrating inky for writer-provided dialogue, allowing branching, NPC knowledge flags, alternate starts, multiple endings, a variety of possible rewards, a ridiculous number of possible objective types, and not once has a quest bug resulted in lost player progression or reward (83 quests and counting)
- Worked extensively with multiple writers to bring their work into the game and implement it as quests within the aforementioned quest system, providing assistance and taking feedback for improvements
- Built a bulletin board system to offer bite-sized quests (re-using much of the logic from the main quest system, but with significantly different player interactions)
- Other contributions include the keyboard/mouse rebinding system, the tutorial system, the notification system, and a variety of other stuff that just sort of happens over the years

Other Work Summary

My non-game work mostly involves a lot of web backend work, both on small singular applications, but also multiple large distributed applications. This includes API development, UI development, testing, observability, and security. A more detailed resume is available upon request.

Other Work Experience

Senior Software Engineer – SmartThings – January 2019 - January 2020

Worked in core infrastructure team, building microservices and diagnosing infrastructural issues

Senior Software Engineer (Contract) – Novus Media – May 2018 - December 2018

Worked to migrate team to modern technologies (Backend, UI, Infrastructure, Mentoring)

Enterprise Architect – Lift Brands – May 2016 - May 2018

Worked to grow team and guided RESTful API designs (Backend, Security, Infrastructure, Mentoring)

Software Engineer – Zags – October 2015 - May 2016

Worked on insurance claim processing software with an emphasis on document generation

Programmer – Saturn Systems – June 2013 - June 2015

Worked on multiple staff augmentation projects building web applications (UI, Testing, Backend)

.NET Developer – United Healthcare – October 2011 - May 2013

Worked on automating data entry tasks including bulk ingest and per-spreadsheet automation

Significant Personal Projects

ECS Experiment – ~2 weeks of spare time – August 2024

An experiment with an ECS-based game “engine” focused on memory contiguity

- Written in C-style C++, focusing on keeping state and logic separate as a matter of thinking clearly about how memory is being laid out
- Component state supplied to systems in such a way as to minimize branching in the inner loop and to enable usage of SIMD intrinsics with zero preparation of the memory being operated on
- Memory segmented by “archetype” (i.e. collections of entities with all the same components)
- Using a third-party task scheduler to split work across multiple threads. Currently, work is split by archetype, with task ordering based on what components systems are defined to read/write
- Integrates the Tracy profiler so I can track how performance changes as I continue the project

Identity Provider – ~1 year intermittent development? – Ongoing

An implementation of the OAuth and OIDC specifications with internal accounts

- A modern .NET implementation of the OAuth 2.1 (draft) and OIDC specifications
  - Supports authorization code flow, client credentials auth, device code auth, resource owner credentials auth (disabled on recommendation of the OAuth 2.1 draft), and refresh tokens
  - JWT/Opaque tokens, PKCE, multiple client auth methods, frontend and backend logout
  - Currently supports TOTP for MFA with FIDO2/WebAuthN support underway
- Minimal frontend dependencies: just bootstrap and popper
- Accounts stored in PostgreSQL and instance coordination via memcached (for account lockouts, etc.)

3AKM LAN Party Site – ~7 months of spare-time development – ~2014

Hobbyist site with LDAP-driven user accounts and database-driven content

- Node.js back-end connected to a MongoDB instance
  - On startup, runs a Browserify bundling routine on client logic and minifies the resulting Javascript for delivery
  - Communicates with an Active Directory instance over LDAP to manage user accounts and permissions (an externally provided requirement), synchronizes user information with the application database for ease of relation to site content (directory remaining authoritative)
  - Heavy use of the bluebird library for promises to handle asynchronous operations
  - Automation of third-party applications relevant to our users (currently only one application, with interactions handled via a UDP socket interface)
- Separate user interfaces for the public site and an administrative portal both written using AngularJS with Bootstrap as a styling basis
- User registration process with reCAPTCHA and email verification

Other Notable Tools/Libraries

Qt, SDL, OGRE3D, Squirrel, OpenAL, TPL, LINQ, Jenkins, Azure DevOps, Git, Subversion, Visual Studio, GCC, Clang, CMake, Node.js, AngularJS, jQuery, Bootstrap, 3DS Max, Photoshop, Unity, Torque3D, Torque2D, Unreal 2.5, Linux, Docker, Kubernetes, ELK, Apache, MSSQL, MySQL/MariaDB, PostgreSQL, MongoDB, Cassandra, Memcached, WinForms, WPF, XAML, MVC, MVVM, HTML, CSS