

SKILLSET

- Expert C++ and C# programmer
- Complex software architecture, large scale problem decomposition, analysis, and research
- Game development with Unity and custom game engines
- Linear algebra and 3D math
- Asynchronous programming and multithreaded data management
- Tools and windows applications using WPF and WinForms
- Relational database design with MySQL, MSSQL, and Entity Framework
- Web development with JavaScript, CSS, jQuery, React, Bootstrap, PHP, NancyFX, and ASP.NET

EXPERIENCE

Freelance Software Developer, Garvey Software

2017 – Present

Custom tailored software design, implementation, testing, and deployment. Key projects include:

Black Seven Studios: Programmed gameplay mechanics in Unity for *Kingdom of Night*. Developed an isometric pixel accurate combat system, player class and skill system, 2D character animation controllers, remappable player input, enemy AI mechanics, scene management, and cutscene logic. Contributed to the successful Kickstarter campaign and ongoing development effort, slated to ship in 2020 on Switch, PS4, Xbox One, and PC.

Modern Round: Added support for stand alone games to an existing mini-game framework. Implemented loading overlay, state, and process management for the game launcher shell. Implemented interprocess communication and state synchronization using Apache Thrift framework and the native Winsock library.

Echelon Technologies: Designed, from the ground up, a web based data entry platform for medical billing information and patient life care plans. Developed front end UI using React and Bootstrap featuring an administrator management interface, unified data search system, data entry UI, and auto calculation logic with client and server side validation. Developed a custom PHP backend CRUD API, authentication system, search system, as well as a life care plan report generator, all interfacing with MySQL as the backing data store.

Anim Tree Editor - Unity Asset Store: Wrote an editor extension aimed at simplifying the process of building animator state machines and blend trees in Unity. Added time saving enhancements missing from the built in Animator toolset including a searchable tree view UI overview, duplicate, disconnect, and drag-drop actions for blend trees, a blend tree template system, and full undo-redo support. Asset store page: <http://u3d.as/1Jza>

Software Developer IV, VirTra

2014 – 2019

Driving Simulator: Led a small team developing a large scale, 4+ screen, police driving simulator in Unity. Responsible for technical roadmap, sprint planning, and mentoring. Designed service architecture for coordinating simulation server, view clients, authoring station, instructor control station, streaming video replay, and hardware control. Developed custom native C++ UDP and TCP networking library for memory efficient, low latency server/client synchronization, service communication, and hardware I/O in Unity. Wrote advanced AI traffic mechanics including vehicle steering, speed management, route planning, lane changes, collision avoidance, and right of way management. Integrated Vehicle Physics Pro package and added dynamic physics simulation LOD, scaling the fidelity of AI vehicle physics based on distance from the user. Implemented custom Unity Editor tooling for road network graph layout and A* path finding. Built a WPF based scenario authoring tool framework with embedded Unity window for world visualization, prop placement, and pursuit path authoring. Performance profiling for garbage generation, CPU, and memory optimization. Designed custom PBR shaders using Shader Forge.

Modern Round: Engineered a framework for developing shooting mini-games in Unity. Implemented a shell application framework that would import mini-game bundles, handle common UI, and control mini-game lifecycle. Supported 3rd party mini-games built with the Modern Round framework and designed the client/server API implemented by 3rd party server developers. Setup and maintained a Jenkins build server farm. Completed various gameplay programming tasks for the suite of 30 mini-games in initial launch. Ported Video Sim and V-Author simulators from native C++ into the Unity based Modern Round framework.

Video Decoder: Created a high performance C++ multithreaded video and audio decoder built on top of libavcodec. Accomplished 60 fps 1080p video playback, and 30 millisecond maximum seek time to arbitrary stream locations. Built as a WPF control and Unity native plugin for use by VirTra's Video Simulator technologies.

VirTra Operator Station: Designed and implemented a keyword searchable web based help system featuring a WPF authoring UI, NancyFX host server, and Apache Lucene.Net search framework. Developed the low light simulation overlay, integrating physical flashlights into simulations utilizing a blob detection algorithm implemented as a C# wrapper over a native C++ Direct Show filter graph reading camera input. Inherited VirTra system services shell and extended it to accommodate WCF based asynchronous service communication. Added a number of core VirTra services including simulation overlay drawing, data collection service, weapon calibration, sound effects service based on NAudio, scenario search tag system, and configuration backup. Created a tablet based remote control application allowing instructor mobility. Performed research and development work for VBS3 simulator integration.

Game Programmer, 2XL Games

2011 – 2014

Capital City: Worked with C++, C#, GLSL, and TSQL as a client and server programmer. Developed post effect rendering techniques, including Gaussian blur, tilt shift, full screen desaturation, and color highlights on user controlled vehicles. Built a shader LOD system for runtime optimization. Created a batch based billboard system for drawing 3D UI elements and foliage interleaved with other semi-transparent game world entities. Wrote camera controllers for touch based map viewing, third person, first person, and idle orbiting. Procedurally animated smoothly blended camera transitions. Implemented A* path finding for AI and multiplayer vehicle navigation. Developed server side statistics, user management, friends, and leaderboard systems using Entity Framework and MSSQL. Designed game match service responsible for asynchronously tracking and enforcing gameplay logic for thousands of concurrent matches.

Spelltorn: Architected service oriented backend server framework, used concurrently across several development efforts at 2XL. Programmed the service host supporting runtime reloading of multiple WCF services configured to run in a sandboxed application domain or as separate process. Designed caching and resource lock management strategies for data access in massively parallel service requests. Researched and developed a lockless service side database key generation scheme using a time stamp, shard, and counter approach, resulting in the ability to generate millions of globally unique keys per second across potentially thousands of services while maintaining rough time sortability.

XLR8: Led the development effort through to app store submission. Implemented UI features, server based user profiles, and store transactions. Tested functionality and iterated on design based on analytics data and user feedback.

Ricky Carmichael's Motocross Matchup Pro: Ported *RCMM* from iOS to PC, Google Play, and Amazon Appstore. Put into place anti-piracy measures, content encryption, Paypal and Amazon AWS payment processors. Performed server maintenance and deployed several bug fix and content patches over the game's lifecycle.

Associate Programmer, THQ Digital Studios Phoenix (formerly Rainbow Studios)

2010 – 2011

Saints Row: Money Shot: Collaborated closely with designers, artists, audio engineers, and other programmers to quickly develop features from prototype through to Xbox Live Arcade and PlayStation Network submission. Used C++, C#, LUA, and Action Script programming languages to tackle a variety of tasks including physics collision response, camera collision, prototypes for aiming mechanics, controller tuning parameters, Scaleform based HUD notifications, goal tracking, FMOD audio hooks, visual scripting, and editor tools all under considerable time constraints.

Dood's Big Adventure: Worked as an engineering intern on a launch title for the Wii *uDraw* game tablet. Improved the toolset, implemented TRCs, and fixed bugs. Profiled and reduced heap allocations to fit the tight memory budget.

Software Developer, Axosoft

2010

Developed features for OnTime using C#, ASP.NET, JavaScript, YUI, and jQuery. Contributed to the version 10.1 patch of OnTime, several upgrades for the online store, and a key activation and verification system for Rocket SVN Server. Prototyped features and UI designs that would be used in subsequent versions of OnTime.

Game Developer, Phoenix Game Studios

2009

Co-founded Phoenix Game Studios, a student run development studio backed by a grant from the ASU Edson Student Entrepreneurship Initiative. Led the three person development team from concept to launch of Ninja Guardian, a vertical platformer for Xbox Live Indie Games written in C# using the XNA framework.

Application Developer, Lab Tech and Advanced Technologies

2007 – 2009

Built applications in C# for test automation, automated test data analysis, graphing, and work-order creation. Interfaced with MSSQL Server to provide a centralized data repository and workflow consistency.

EDUCATION

- B.S. Computer Science, Arizona State University, 2010
- Certificate in Computer Game Programming
- Computer aided geometric design and rendering coursework