

Michael Fritz

Software Engineer

(360) 355-4365 | michael.s.fritz@gmail.com | www.linkedin.com/in/michaelscottfritz

Technical Skills

Concepts	Languages	Tools	Libraries / Protocols
- Algorithms and Data structures	- C	- Git	- OpenGL 3
- Operating systems	- C++	- Perforce	- DirectX 11
- Linear algebra	- C#	- SVN	- Berkeley socket API /
- Geometry	- Javascript	- Make	Winsock2
- Network programming	- Python	- Windows	- UDP / TCP / IP
- Graphics programming	- ARM assembly	- Linux	- Win32
- Physics programming		- Vim	- Unity
- Game engine development			

Professional Experience

Game Engineer Intern

May 2020 - Aug 2020

AuthorDigital

- Built game systems alongside a team of 8 using Agile methodologies.

Game Project Class TA

Feb 2020 - Dec 2020

DigiPen Institute of Technology

- Mentored student teams developing 1 year long game projects.

Education

BS in Computer Science in Real-Time Interactive Simulation

Expected: Apr 2021

DigiPen Institute of Technology

3.88 / 4.0 GPA

AA in Computer Science

2015 - 2017

Centralia Community College (Attended during high school)

3.98 / 4.0 GPA

Projects

Gameplay Programmer

Sep 2018 - May 2020

DigiPen Game Projects

Team of 12-15

- Researched Delaunay Triangulation and the Voronoi Diagram through scholarly articles.
- Implemented a pipeline using the Bowyer-Watson algorithm and Voronoi Diagram clipping to break meshes.
- Tested the algorithms using a debug pipeline to ensure reliability.
- Developed the architecture and framework for the custom 2D engine in C/C++ to support 6 programmers.
- Designed an ECS framework from scratch to allow for growing functionality while promoting CPU cache efficiency.
- Built a robust collision detection and resolution system for the game's ball to prevent tunneling at extreme speeds.

Graphics Programmer

Jun 2019 - Sep 2019

Handmade Software Renderer - 3D Graphics Engine

Personal Project

- Created a 3D software renderer from scratch in C/C++ to learn more about the graphics pipeline.
- Optimized triangle rasterization using incremental point-triangle collision in order to render in real time.
- Solved accurate clipping to the NDC cube in clip space to robustly render clipped triangles.

Graphics Programmer

Jun 2018 - Sep 2018

Particle System Program

Personal Project

- Created a visual effects program in C/C++ using OpenGL 3 to simulate tons of particles.
- Optimized the system using multithreading and compute shaders to add 10+ million particles at 60 FPS.