Grayson Clark

Email: gclark1013@gmail.com
Portfolio: faultypine.github.io
Mobile: +1 847-951-1965
Github: github.com/FaultyPine
LinkedIn
Education: Indiana University - BS in Computer Science August 2020 - July 2024

Experience

- **Blizzard Entertainment - Diablo 4** — In-Person - Summer Internship
  Engine Programmer — *May 2023 - August 2023*
  - **Improved Artist Workflows**: Improved engine tooling based on Artist/Designer requests to reduce iteration times and friction
  - **Bug Squashing**: Closed a total of 23 Jira tickets throughout the summer. Worked with QA, designers, and other programmers
  - **Documentation**: Wrote a "Codebase Architecture Overview" document to help get new engineers familiar with the codebase. Also wrote documentation for a proprietary Blizzard library
  - **Live++**: Integrated the C/C++ Hot-Reload tool Live++ into the Diablo 4 proprietary engine. Reduced average programmer iteration cycle from avg. 5 minutes for compile-link-reboot to <10 seconds.
- **ThirdPixel Interactive - Smack Studio** — Remote - Contractual
  Netcode Consultant — *Febuary 2022 - May 2023*
  - **Codebase Analysis**: Analyzed the team's codebase to provide advice about recommended project architecture
- **Indiana University Bloomington** — In-Person
  Undergraduate Instructor - Introduction To Computers and Programming — *August 2022 - Present*
  - **Team Meetings and Grading**: Created assignments, graded and proofread student code
- **Deerfield Park District** — In-Person - Seasonal
  Elementary School Summer Camp Counselor — *May 2015 - July 2021*

Projects

- **Brawlback - multiplayer client (Deterministic Lockstep, Speculative Execution, Client Synchronization, Peer-To-Peer Networking)**: (Work in Progress) Peer-To-Peer Networking client for Super Smash Bros Brawl with Integrated Matchmaking that allows players to instantly play with each other even in volatile or poor network conditions. Written in C++ and PowerPC Assembly
- **Tiny Engine - simple game and game engine/framework (C++, OpenGL)**: Created a lightweight framework of APIs wrapping OpenGL and GLFW for videogames programming. Features include hardware-accelerated rendering, debug shape drawing, spritesheet parsing/animation, 3D mesh (.obj) loading/rendering, shader API, postprocessing, and a simple job system for easy multithreading.
- **HewDraw Remix - overhaul modification (Rust, Code Injection)**: (Formerly lead developer) open source modification of Super Smash Bros Ultimate. Major contributions included a full rewrite of the codebase from C++ to Rust in a 2 month period, a Continuous Deployment system, and a user-facing launcher/updater application. Currently we have around 25 contributors and 4000 active players. The project is composed of approximately 100,000 lines of Rust, as well as python scripts for automating various tasks and infrastructure
- **UltimateModShop - User-facing shop/management app (GUI, Web API, Embedded Software)**: a homebrew application written in C++ for the Nintendo Switch that allows users to download, install, and manage Smash Ultimate mods directly on their switch, using the GameBanana web API.
- **Reverse Engineering / Skyline Plugins (Embedded Software, Reverse Engineering, Code Injection)**: Utilized deep knowledge of C++/Assembly and an intuition for common code practices to Reverse Engineer parts of the Super Smash Bros Brawl and Super Smash Bros Ultimate executables. Interfaced with the Nintendo Switch's developer SDK to perform file IO, networking, etc. Used Reverse Engineered code and the Skyline library to properly perform code injections and overwrite game behavior with our desired behavior. Written in Rust. Created some small projects and was previously lead developer on HewDraw Remix
- **Wave Function Collapse Algorithm Visualizer (Algorithms, Data Structures)**: Implemented the Wave Function Collapse algorithm in JavaScript
- **Unity Projects - game development (Unity, Shaders, Rendering)**: Created various Unity projects that recreate game mechanics from popular games. Includes Cel Shading, Particle Systems, UI Integration, Image Effect Shaders, Water Shaders
- **Solary - game development (Unity, Shaders, Procedural Generation)**: Created a physically accurate solar system simulation with procedurally generated planets.

Skills Summary

- **Languages**: C++, Python, C, Rust, C#, HTML/CSS, Java, JavaScript
- **Platforms**: Windows, Linux, Web, Raspberry
- **Skills/Tools**: GIT, GDB, WinDbg, WSL, Valgrind, GCC/G++, MSVC, Unit Testing, CI/CD, Perforce, Jira, Confluence, Ghidra, Visual Studio
- **Soft Skills**: Communication, Organization, Time Management, Conflict Resolution, Cooperation