

Nick DaCosta

Software Engineer

Swanton, VT | 973-885-0830 | DaCosta.Nick@yahoo.com | linkedin.com/in/nick-dacosta | dacuster.dev

SUMMARY

C/C++ software engineer with experience building cross-platform SDKs, developer tooling, and CI/CD build automation systems, plus a background in skilled trades. Seeking tools/build engineering roles in games, simulation, robotics, or embedded systems.

TECHNICAL SKILLS

Languages: C/C++, C#, Bash, JavaScript, JSON, Python, OpenGL, SQL

Build & Automation: CI/CD, CMake, Jenkins, TeamCity, GitHub/Gitea Actions, vcpkg

Version Control: Git, SVN, Perforce, Gitea, GitHub

Systems & DevOps: Docker, AWS, Apache (reverse proxy), Cloudflare, PostgreSQL, OAuth SSO, Linux (Ubuntu), Windows

Frameworks: CLion, Unity, Rider, Qt, Dear ImGui, Visual Studio, Unreal Engine

WORK EXPERIENCE

Software Engineer II

Apr 2025 – Mar 2026

MicroStrain by HBK (formerly Parker-LORD), Williston, VT

Responsible for maintaining the inertial sensor APIs for the [MIP SDK](#), managing build and automation systems for the software team (extending to the embedded engineering department), and providing inertial software customer support.

- Overhauled the [MIP SDK's example projects](#) — including their documentation, logging, and error handling — as sole engineer, standardizing C/C++ "getting started" examples and earning positive feedback from 10+ customers.
- Led migration of CI/CD pipelines from Jenkins to GitHub Actions, consolidating shared logic into reusable workflows to reduce per-project configuration overhead.
- Refactored [MSCL](#) and SensorConnect dependency management using CMake and vcpkg, cutting new developer onboarding time from multiple days to under 30 minutes.

Software Engineer I

Oct 2021 – Apr 2025

MicroStrain by HBK (formerly Parker-LORD), Williston, VT

- Developed and maintained MSCL, a C++ sensor communication API with Python/C# bindings via SWIG, built via a CMake-based Jenkins CI/CD pipeline.
- Led development of the MIP SDK, a C/C++ embedded successor to MSCL, designing a minimal cross-platform API adopted across multiple embedded applications.
- Investigated a Microsoft WebView layering limitation in InertialConnect, a 3+ year C++/DearImGui desktop app, reverse-engineering Win32, Wayland, and X11 windowing internals.
- Redesigned the InertialConnect rendering framework to be windowing-library agnostic, decoupling OpenGL/DearImGui from SDL2 and enabling swapping with GLFW, eliminating unmaintainable forks.
- Architected a reusable, templated sensor-configuration widget system for InertialConnect, reducing new widget development time by ~90%.

Advanced Programmer

Sep 2020 – Jun 2021

Champlain College, Burlington, VT

Wrote gameplay and tooling scripts for Unity-based student game projects, including networked multiplayer and remote database integration.

Electrician Assistant <i>Champlain College, Burlington, VT</i>	Aug 2017 – Aug 2020
Handyman <i>DaCosta Contractors, Warren, NJ</i>	Jan 2015 – Jun 2017
HVAC Technician <i>H.V.A.C. Mechanical Contractors Inc., Union City, NJ</i>	Feb 2011 – Oct 2014

EDUCATION

DePaul University, Chicago, IL (Online) Master of Science in Game Programming – Enrolled Part-Time	Aug 2022 – Mar 2024
Champlain College, Burlington, VT Bachelor of Science in Game Programming, Minor in Data Science	May 2021
Raritan Valley Community College, Branchburg, NJ Game Development	Jan 2016 – May 2017
Lincoln Technical Institute, Mahwah, NJ Heating, Ventilation, Air Conditioning, and Refrigeration	Feb 2013

PROJECTS

Through Rust We Are Returned – Tools Programmer Shipped Title: free turn-based tactics RPG released on Steam as senior capstone (Chaos Crew Productions).	Jan 2021 – May 2021
---	---------------------