**SKILLS & SPECIALTIES**

***Languages****:* C++, Python, C#, Java, ActionScript 3, Lua, etc.

***Platforms****:* PC, Nintendo Switch, Steam, PS4, XboxOne, iOS, Android, Nintendo DS, Linux (limited)

***Engines****:* Unreal 4&5, CryEngine, Godot (limited), Gameloft's Glitch Engine (Irrlicht based), Unity 3D (limited)

***APIs****:* WinAPI, WinForms, Win/POSIX Threads, Berkeley Sockets, DirectX 9/10 & HLSL, OpenGL 2/ES

***Unreal Engine:*** Animation Blueprints, Animation State Machines, Blend Spaces, Blueprints, Common UI, Data Assets & Tables, Enhanced Input, Gameplay Ability System, Gameplay Debugger, Material Shaders, Montages, Networking, Unreal Insights, etc.

**EMPLOYMENT HISTORY**

**Singularity 6,** Senior Gameplay Engineer **Apr 2019 – Apr 2024**

***Palia*** (PC, Switch, Steam)

* Cozy multiplayer community game in Unreal 4 and 5 with C++, Blueprint, and Python, using UE’s Gameplay Ability System
* With Animation lead set foundation for character’s locomotion and tool handling, and how fish animate in the minigame
* Wrote RFCs to: set the foundation for creating cheats and debug tools for every gameplay feature; set the foundation for feature creators to create and maintain feature test maps; Animation Blueprint best practices
* Wrote docs on: Actor & Component Replication guarantees to improve upon private Unreal docs around it; Montage gotchas; Gameplay Debugger use; cheats for every gameplay feature and accompanied UI cheats, console variables/commands, and Gameplay Debuggers for said feature
* Created productivity enhancing build and launch scripts
* Tiered granular cheat detection – currently only utilized by the Gardening skill
* Worked on most of my team’s systems, player skills, player components, player abilities, UI, generic interactor and interactable system, gatherables & shops, player & NPC movement optimization, telemetry, etc.
* Added first accessibility feature: 3 different sprint modes so players can choose their preferred mode

**Treyarch Studios (Activision),** Gameplay Engineer **Mar 2017 – Apr 2019**

***Call of Duty: Black Ops 4*** (PC, PS4, XboxOne)

* Worked with C++, game script, and Lua
* Developed weapons, gadgets, and score streaks
* Single engineer who guided designers/scripters with Combat Training
* Fixed script serverfield system, Script VM bugs
* Worked on player abilities, created three minigames, vehicle flare for impacting incoming missiles, flying vehicle roll with fancy cam, vehicle AI behavior, vehicle tail/brake/reverse lights
* Made material physics affect vehicles, friction and fluid depth resistance (water, mud, sand, etc.), and hydroplaning

**Blind Squirrel Games,** Gameplay Programmer **Mar 2016 – Feb 2017**

***Trove*** (PS4, XboxOne)

* Helped port PC MMO, Trove, to PS4 and XBox One
* Improved input action system and made data driven bindings, fixed controller issues, camera issues, etc.
* Fixed tricky memory violations and compute shader issues
* Saved UI weeks of work, designers were so happy they hugged me

**Red 5 Studios,** Gameplay Programmer **Oct 2015 – Feb 2016**

***Firefall*** (PC)

* Prepped Firefall for the Chinese market and helped port x64 client
* Worked on cool new features like the time dilation bubble

**IllFonic,** Gameplay Programmer **Nov 2013 – Sept 2015**

***Star Citizen*** (PC)

* Created magnetic boots for walking around arbitrary surfaces (Crytek praised)
* Handled player rotation and movement, including making CryEngine support zero-gravity player movement
* Set up all weapons and gadgets; revamped weapon sway and recoil
* Created grapple beam gadget with impulses to cancel extra velocity, speed player toward walls, and slow before impact
* Created IK based weapon sway, and procedural helmet sway to replace animation-based movement

**Gameloft,** Generalist Programmer **Dec 2011 – Oct 2013**

***Cosmic Colony*** (iOS), ***Ice Age Village*** (iOS), ***Oregon Trail American Settlers*** (iOS)

* Lead prototype, created Python data pipeline that efficiently handled gigabytes of data
* Performance improvements, dynamic loading/unloading sprites, fixed memory leaks, created monetized mini-game, built/uploaded DLC, managed a DLC release, did QA and Gold builds for iOS App store, etc.
* Worked on Python scripts and C# tools that made once tedious work quickly manageable

**1st Playable Productions,** Gameplay Programmer (Co-Op) **Jun 2009 – Nov 2009**

***World of Zoo*** (NDS), ***Club Penguin: Herbert's Revenge*** (NDS)

* Created mini-games in C++ for World of Zoo (pictured on the back cover), and for Club Penguin
* Setup story sequences, puzzle items, & created scripting queue system for Club Penguin, etc.

**FREELANCE**

**Underminer Studios,** VR Contractor (Part-time) **Jan 2016 – Jul 2016**

***Vrideo*** *(PC, PS4)*

* Created Unity C++ HTTP video streaming plugin with FFmpeg for Win32 and made sure it worked flawlessly with HTC Vive and Oculus Rift. Created 60fps Unity C++ HTTP video streaming plugin for PS4

**PROJECTS**

**Distorted Realities**, Gameplay Programmer (Indie) **Present**

***[Untitled greybox prototype]*** (PC, Steam Deck, Linux)

* C++ and GDScript. Experimentally pushing the limits of Godot Engine. Diving into AI, aiming for F.E.A.R. and Left 4 Dead
* Singleplayer FPS trainer, help player improve their weapon handling skills and make them feel like a GOD

**EDUCATION**

**Rochester Institute of Technology,** Bachelor of Science, Cum Laude (Video Game Design & Development) **2011**

**ACADEMIC PROJECTS (Select)**

***2D PlasmaTech Engine***, Engine and Game Systems Programmer **2010 – 2011**

* C++ with DirectX 10. Setup rendering systems: Effect file, Sprite Sheet, Camera, Mesh, etc
* Wrote Content Manager for loading textures and models. Added Math-helpers (ray picking screen-to-world, etc.)

***Upwards*** (open world voxel game in sky), Engine Programmer **2010**

* C++ with DirectX 9. Start of my *cross-platform* Warp Engine w/ components abstracting platform functionality
* From scratch: UI systems, gadgets, font renderer with Unicode code points, etc.

***Robo Assault***, (3rd Person Networked Shooter), Gameplay Programmer **2010**

* C# with XNA. Systems creation: player camera & controls, player physics, networking, robust kill-feed messaging, etc.

***Fluffymancer & unnamed RTS***, (3rd Person Shooter & Top down RTS), Gameplay Programmer **2008 – 2009**

* C# with XNA
* Systems creation: NPC AI, spatial store, skinned animation, debug primitive batch drawing, etc.