

SUMMARY OF QUALIFICATIONS

I am a Technical Artist with over ten years of industry experience on both AAA and mobile titles. I specialize in lighting and art pipeline optimization, but also have experience with shaders, visual effects, tools development, and prototyping

WORK EXPERIENCE

Undead Labs, Senior Technical Artist
"State of Decay 2", Xbox One / PC

June 2012 to Present

- Lighting. I authored all lighting, skies, and post processing for State of Decay 2. I worked closely with internal engineers and external middleware vendors to get the lighting and the sky to meet both visual and performance goals. I also developed an asset pipeline to allow assets to automatically import using correct lighting settings.
- Photogrammetry. I developed a photogrammetry pipeline for scanning human faces into State of Decay. I built my own photogrammetry rig, scanned over 150 people, processed the scans into high res models, and passed them off to our character artist. This had a significant increase to both quality and production speed on our human characters.
- Master Shader Library. I created a master shader library and worked with another Technical Artist to develop a Photoshop export pipeline that efficiently packs textures and speeds up the export process. This library uses Material Instances to reduce compiled shader permutations, save memory, and allow for global control over shared material parameters.
- Vehicle Deformation System. I designed, prototyped, and wrote the exporter and shader for State of Decay 2's vehicle damage system. This system baked blend shapes into vertex colors to store vertex deformation and normal deviation and provides masking for the corresponding blended material.
- Real-time Time of Day. I created the system that controls visual time of day throughout the game. This drives the sky, the lighting, the post processing, and ties into the game's time of day system for gameplay events and audio.

"Moonrise", iOS / Android / PC

- Lighting and Post Processing. I handled all lighting, both static and dynamic, for all 35 environments in the game. I also worked closely with another TA to make sure the character shader ran efficiently with our lighting model.
- Weather System. Moonrise had a very limited number of environments. To increase variety, I used lighting and effects to create four different Time of Day / Weather variations of each environment. This prevented players from getting bored by seeing the same environments over and over again and went a long way to make the world feel more alive. This cost us very little, both in disc space and time.
- Character Combiner. I designed and created the system for combining dynamic character pieces into a single character rig at runtime. This had a huge impact on performance.
- Dynamic Spell Icons. I designed a system and wrote the shader for dynamically coloring UI icons. Recoloring and reusing icons drastically reduced the number of UI textures needed for the game.
- Dynamic Portrait Capture System. Moonrise characters had hundreds of customizable options, and we needed the character's 2D portrait to match the models. I created a system that captured the character's portrait with preset lighting into a RenderTexture whenever a character was instantiated.
- Memory Profiling and Asset Policing. I routinely collected asset stats and helped artists make assets as efficient as possible.
- UI Implementation. Moonrise was a UI heavy game, and we were short-handed in the UI department. To help get the project on schedule, I spent several months building UI prefabs out of Photoshop mock-ups and passing them off to programmers to implement functionality while we searched for another UI artist.
- Shaders. I wrote several cg shaders that were used across the project. These included shaders for UI elements and some environment shaders that tiled textures in world space.

WORK EXPERIENCE (CONTINUED)

"State of Decay", Xbox 360 / PC

- Interior Light Volumes. Crytek, the engine used for State of Decay, was ill equipped to handle rapid transitions between interior and exterior spaces while using a dynamic time of day system. In order to provide the illusion of recession over space, I created a system of negative lights and volumes to pull out sunlight and provide light falloff from surfaces as they get farther from windows.
- Buildings. We outsourced most of our buildings to a third party, and the assets we got back were about 70% game ready. I handled the remaining 30% to get the buildings up to standard. This included geometry polish, cleanup, and tessellation, vertex light bakes, texture rework, and physics material volumes.
- Memory Profiling, Asset Policing, Asset Reduction. SoD was an open world game with no loading screens in which players could enter any building at any time. Special focus was put on maintaining visual quality that was up to industry standard while still maintaining asset density that made the world feel "lived in". We had to strike a very delicate balance between performance and quality, and I was in the front line of that particular battlefield.
- Asset Database. I worked with our Technical Art Director to co-author a SQL database and interface for tracking asset status. This allowed us to quickly search through the properties of all assets in our library and, for example, find models that did not have physics geometry.

Monolith Productions, Technical Artist / FX Artist

Dec 2009 to May 2012

"Guardians of Middle Earth", Xbox 360 / PC

- FX Artist. After doing a lot of tools work on the previous project, I wanted to get back to creating art, so I transferred over to the FX art team. Over the year and a half I spent on this MOBA style game, I created effects for almost a dozen champions, worked closely with the other FX artists to unify the art style, trained several other artists on our tools, and worked with engineers to improve the FX pipeline for our in-house engine.
- Destructibles. I built the pipeline for creating destructible objects.
- Training. Monolith used a proprietary engine, and none of it was documented. Our effects team grew from 0.5 people to 6 people in 2 months, and I was in charge of training the new artists on the effects tools.
- Engine Improvements. Guardians of Middle Earth was a lot more effects-heavy than any game Monolith had previously done, and the FX tools were not expansive enough to handle the demands of the project. I collected tool requests from myself and the rest of the FX artists and worked with the engineering lead to prioritize and implement tool requests.

[Cancelled Project], Xbox 360 / PC

- The Architect. I co-authored a building creation tool that allowed us to rapidly build metropolitan buildings out of small assets. I handled design specs, test assets, and UX.
- Road Tool. I prototyped a road creation system in 3DS Max to rapidly create city layouts. This system used nodes and splines to quickly lay out an entire city. It handled roads, sidewalks, intersections, crosswalks, and arbitrary changes in elevation.
- Cloth Physics. Monolith's art team had no experience with the Havok cloth plugin for 3DS Max. Having previous experience with Havok in 3DS Max at Surreal, I established and documented the workflow for creating cloth assets for environmental objects such as flags, tarps, etc.

WORK EXPERIENCE (CONTINUED)

Surreal Software, Associate Technical Artist

March 2008 to Dec 2009

"This is Vegas", Xbox 360 / PS4 / PC

- **LOD Tool.** I played a large role in designing a system for projecting textures from game res assets to low poly LOD for an entire city block. Rather than individual objects having LODs, we created one giant LOD for an entire city block. This significantly reduced object count as opposed to per-object LODs. It also dramatically improved streaming, as object count was our biggest bottleneck. I profiled, established, and managed texture resolution, mesh density, and memory usage for all LODs to ensure best possible quality without sacrificing the quality of art for the game. Once we were satisfied with the system, I modeled, unwrapped, and baked the LODs for 148 city blocks worth of content.
- **Shader Complexity Tool.** I proposed, oversaw, and utilized an in-game view mode that visualized shader complexity. This blue-green-yellow-red heat map shaded the world based on the number of shader instructions of each material. This tool had a significant impact on our ability to quickly identify and target needlessly expensive content.
- **Vehicle Physics.** The person responsible for setting up the vehicle physics left the company shortly before I arrived at Surreal, and the process for creating physics constraints for vehicles was completely undocumented. Armed with a few half-working files, I had to figure out how to set up Havok physics for our fleet of vehicles, document the process, and later train others on the vehicle setup process.
- **Memory Management.** I routinely gathered and analyzed statistics on memory usage through in-game tools and PIX. I used this data to help artists keep their environments within budget.

Handheld Games, Intern

June 2008 to Dec 2008

"Phinnes and Ferb", "Disney Fairy Princesses", Nintendo DS

- **Nintendo DS titles.** While interning at Handheld, I worked on several Nintendo DS titles. Responsibilities included pixel character art, sprite animations, and background painting.

Self Employed

August 2003 to Dec 2009

[Indie iOS Games]

- To make some money to help me through college, I created and shipped 4 solo indie games for iOS when the platform was in its early stages.

[Interactive Media Designer]

- Designed websites and training videos for several tech hardware companies.

EDUCATION

Bachelor of Fine Arts, Game Art and Design, Art Institute of Seattle

June 2008

Game Art and Design
Graduated with Honors, Deans List

Associates Degree, Fine Arts, Madison Area Technical College

June 2004

Interactive Media and Design

SKILLS / SOFTWARE

Software: Maya, 3D Max, Zbrush, Photoshop, Illustrator, After Effects

Programming: C#, Javascript, Python, Maxscript, CG

Engines: Unity, Crytek, Unreal 3&4, LithTech

Platforms: iOS, Android, Xbox One, Xbox 360, PS3, Windows