

LOREN D. HOFFMAN

(509) 338-5517
Loren.Hoffman@gmail.com

12907 SE 277th St
Kent, WA 98030

PROFESSIONAL EXPERIENCE

Portalarium, Austin, TX

Jul 2012 – Apr 2013

Rendering Engineer

- Designed and developed game systems for Unity3D independently, and with team members
- Developed functionality across desktop and mobile platforms while maintaining quality and performance
- Designed and built real-time hex based terrain generation and streaming system
- Created a dynamic environment skybox with day-night transitions, clouds, atmospherics, and detail maps
- Built a dynamic weather system supporting cloud cover, color transitions, lightening, and GPU accelerated rain
- Developed the drag-and-drop spatial inventory system and the developer console using NGUI
- Extended NGUI to support distance field fonts and procedural data driven UI structures

Zebra Imaging, Austin, TX

Feb 2012 – Jul 2012

Rendering Engineer

- Worked to fix and bring to a maintainable state a core internal point cloud and digital terrain preparation tool
- Repaired broken systems, extended file format support, fixed performance issues, and improved the UI experience
- Extended point cloud processing systems to better support the flexible data formats
- Documented the original code base design and integrated Doxygen comments
- Prototyped a new point cloud editor complete with a basic UI, multiple format support, and fast performance

Controlled Chaos Media, Dallas, TX

Jan 2011 – Jan 2012

Software Developer, Lead

- Operated as lead developer: developing projects, overseeing projects, creating estimates and designs
- Solo developed an unannounced level editor with procedural geometry generation
- Developed a system to automatically stabilize augmented reality positioning input
- Completed multiple mobile contracts, as company representative interfaced directly with the customer
- Produced technical design documents from customer requirements
- Worked extensively in C++ and Unity3D

PROJECTS

Alchemy Engine

2013

Founder, Developer

Custom Engine

- Created a component driven game engine designed for multi-player role playing games
- Designed and developed a component RPC system with automated end-to-end dispatch
- Built a flexible rendering system with both forward and deferred rendering implementations
- Developed a new voxel modeling tool using Qt with Lua as the extension language
- Implemented exporters for OBJ, IQE, and the engine specific format
- Used boost for ASIO, bind, function, and other systems
- Used CMake for build management and re-build minimization

Shroud of the Avatar - Portalarium

2012-2013

Rendering Engineer

Unity3D

- Designed and developed the overland hexagon map's terrain generation and streaming
- Terrain system supports very large hex grids and can generate in real-time both on desktop and mobile
- Implemented procedural additions of rivers and roads into generated height maps and splats
- Added support for generating and batching vegetation application onto the terrain
- Built an Unity3D editor addition for managing terrain data including textures, tile types, and vegetation
- Developed a dynamic skybox with day-night transitions, nebulae, celestial bodies, clouds, and atmospherics
- Created a GPU accelerated weather system with support for clear skies to rain and cloud cover variations for lighting
- Implemented character sheet and bag system with drag-and-drop items

LOREN D. HOFFMAN

(509) 338-5517
Loren.Hoffman@gmail.com

12907 SE 277th St
Kent, WA 98030

Port Casino Poker - Portalarium

Software Engineer

2012

Unity3D

Primary client developer, worked on all versions: Flash, Web client, iOS, and Android
Developed controller-view driven user interface elements and data driven UI widgets for NGUI
Corrected data flow issues in client and consolidated state storage
Implemented network error management and recovery system to survive mobile network hopping and poor connections
Aided in identifying bugs introduced in Unity3D-to-Flash cross compiler
Worked on platform specific components including Facebook and store integrations

Wizard RPG – Master’s Thesis Project

Developer

2012

Custom Engine

Developed a general purpose component driven game engine
Built a JSON library and factory system for level specification and data driving components
Developed a 2D subtractive physics system for specifying world bounds with additive blocking volumes
Created a deferred renderer with support for emissives, glow, point lights, directional lights, and shadows
Designed and built a path finding system using ray marching for fast path smoothing
Built a small particle editor for tweaking particle settings to save into a component
Created a simple voxel modeling tool for creating game assets

Holographic Definition Tool Federal – Zebra Imaging

Rendering Engineer

2012

Qt and Open Scene Graph

Worked to fix bugs, simplify design, extend features, and improve user experience
Reworked custom UI into re-usable Qt components to reduce code duplication
Fixed scene hierarchy to correct stabilization issues with massive terrain sets
Extended software settings, hotkeys, and automation based on interaction with internal customers
Fixed layouts to be entirely procedural to support more screen resolutions and multi-monitor setups
Worked with QA and production teams continuously to ensure product target was hit

Dallas Arboretum Augmented Reality – Controlled Chaos Media

Core Tech

2011

Unity3D

Developed core augmented reality technology around String and Qualcomm’s systems
Built a stabilization system to enable AR use in low light, sharp angle, long range, and moving scenarios
Wrote a generalized AR layer for working with the various APIs
Helped artists design markers to better work with the different AR detection algorithms

Verticus – Controlled Chaos Media (un-credited)

Core Tech

2011

Unity3D

Developed the core level streaming system used for the infinite runner style game
Built system so that levels could be created as a Bezier spline specified by designers
Designers could specify level colors using a tone system which procedurally affected level geometry
World elements supported manual placing and procedural spawning based on spline ranges
Added arc-length-reinterpretation system to spline library to speed up level logic
Pooling system maximized object re-use and minimized memory thrashing during play

Unannounced Level Editor – Controlled Chaos Media

Lead, Developer

2011

Qt and OpenGL

Specified, designed, and developed a level editor using C++, Qt, and OpenGL
Implemented concave polygon tessellation, 3D subtractive volumes, and robust geometry generation
Designed and developed the menus, panels, object and property managers using Qt
Built an easy to use vector graphics editor mimicking controls found in Adobe Illustrator
Implemented multiple exporters including OBJ and FBX
Managed testing and feedback sessions, feature planning, and schedule estimates

LOREN D. HOFFMAN

(510) 338-5517
Loren.Hoffman@gmail.com

12907 SE 277th St
Kent, WA 98030

INDIVIDUAL DEVELOPMENT

- Real-time Ray Marched Terrain Renderer*** 2010
Developed DirectX 11 compute shader API abstraction
Created procedural terrain renderer by ray marching against a height map using the API abstraction
Optimized ray marching shaders for performance to support 250k rays per frame
- Goo Gun*** 2009
Created a real-time “goo” simulation
Simulated the interactions of tens of thousands of particles on the CPU per frame using Verlet integration
Utilized constraints and springs between particles to simulate the interactions of the goo balls
Designed the system to support many inputs such as the breaking point of connections and spring strength
Rendered goo by voxelizing the data and rendering the voxels using marching cubes
Used profiling to optimize the system for CPU side computation
- Terrain Renderer*** 2010
Built a terrain rendering component for personal 3D engine capable of rendering 8k height maps
Utilized an H-fractal based index pattern to optimize usage of the vertex cache
Built a level of detail system that gives preference to terrain more likely to have popping
Designed and built an elegant stitching algorithm using vector translations in a 2D space to compute indices
- ASDF Scripting Language*** 2010
Developed an iterative scripting language with lexical analyzer, parser, compiler, and virtual machine
Built the virtual machine around reference counting to support implicit resource management
Language features, such as slices, borrowed from Python and Lua to maximize utility
Integrated virtual machine into my game engine so function calls could be made into and out of the VM
Compiler tool chain and virtual machine profiled and optimized for speed
- General Engine Development*** 2009-2011
Built API agnostic display driver capable of rendering OpenGL or DirectX 10
Constructed a BSP collision system with information loaded from Quake 3 level files
Built a Quake 3 map renderer with tessellation of the Bezier patches and using PVS data for culling
Created a Collada model format file loader
Developed a 3ds Max animated model exporter and importer pair for my 3D engine
Developed a shadow map pipeline with smoothed shadows using root-mean-squared averaging
Designed and built a JSON library for easy saving and loading of data from mapped classes

EDUCATION

- The Guildhall at Southern Methodist University, Plano, TX** Aug 2009 - Dec 2010
Master of Interactive Technology, specialization in Software Development
- Washington State University, Pullman, WA** Aug 2004 - Aug 2009
Bachelor of Science in Computer Science
-

SKILLS

C/C++, Java, Python, ActionScript, Lua, JavaScript, C#, Objective-C, OpenGL, GLSL, CG, DirectX, HLSL, Unity3D, Flixel, Boost, Qt, Forward Shading, Deferred Shading, TCP, UDP, JSON, Multithreading, Spatial Partitioning Systems, Terrain Systems, Particle Systems, Meshing, Splines, Make, CMake, Windows, Linux, Unix, OSX, SQLite, Visual Studio, Eclipse, XCode, SFML, SDL, Scons, SVN, Git, Mercurial, Perforce, Doxygen, Augmented Reality