Branden Turner

Software Engineer

636·297·5484
San Francisco, CA
brandencturner@gmail.com

Objective: A position as a software engineer writing performant code, either in embedded systems, or tools

Highlights

- Communication is the most crucial part of my development process. I learn by communicating what I don't know, and sharing what I do know.
- Designed and developed a high-profile, unannounced Walt Disney Imagineering project that leverages multiple hardware and software systems working together in a highly performant manner
- Designed and implemented graphics engine, geometry library, 3D pipeline and animation system in 5 weeks
- Implemented development tools for game designers and artists by communicating first-hand with them and gaining iterative feedback, then implementing their requests on a fast timeline

Career History

R&D Imagineer, Walt Disney Imagineering **2/15 – Present**
Working within the Creative Technology Studio alongside ILM and Lucasfilm to bring new experiences into Disney Parks and Resorts, and creating tools and workflows to better implement these experiences. Projects include Star Wars Land, collaborating with ILM's X-Lab, and unannounced projects that mix hardware and software while maintaining performance. Primary languages include C++ and Python.

Software Engineer, Good Mood Creators **9/12 – 2/15**
GMC is a small startup game studio, so I did all sorts of programming including game logic, tools, graphics, animation, and AI. Primary languages included C# and CG Shaders, with some Python and Pymel for tools. Communication was a central part of this job, since it was a small team that needed a lot of custom tools that worked with various software, from game engines to 3D content packages.

Professional Programming Intern, Walt Disney Imagineering **9/13 – 12/13**
Worked on previsualization technology on multiple upcoming projects as part of the Creative Technology Group. Software used includes Unity, Maya, MotionBuilder, Nuke, Motive, Arena, and Panda3D. Tasks included writing, testing, and maintaining software for use with the previously mentioned software packages in Python and C++ as well as MelScript.

Contract Software Engineer, University of Washington **9/12 – 6/13 & 8/14 – 4/15**
Developed a game prototype and website for the University of Washington School of Medicine and Health Sciences. Responsibilities included programming using PHP, JavaScript, MYSQL, and HTML/CSS, as well as game and website design and implementation, and other various technical tasks, including documentation for non-technical users. The second project expanded on the first, and was done in the Meteor framework, using mainly JavaScript with some HTML, CSS, and Handlebars.

SDK Engineering Intern, Havok **6/12 – 9/12**
Implemented new systems and improved existing tools that streamline demo development as well as testing, refactoring and restructuring obsolete code. Using my newly designed systems, I updated over 30 demos for the Behavior product. Primary programming languages were C++ and Lua, with a bit of C# for frontend tools.

Projects

Generalist Programmer **9/12 – 2/15**
Mekazoo: 2.5D Platformer in the Unity Game Engine (Unity 4/5)

- Implemented 3D camera system including trigger volumes, scripted events, and keyframed animations
- Optimized gameplay and rendering for multiple platforms, including PS4, XboxOne, WiiU, and PC
- Designed and implemented creative tools to be used within Maya, 3DsMax, and Unity, working with artists and designers for iterative feedback
- Co-architected and implemented gameplay systems and multiple character controllers (C#)
- Responsible for debugging a large portion of the major issues we encountered

Pipeline Architect / Graphics Programmer / Producer **8/11 – 4/12**
Grimelins: (C++, DirectX 9) A single-player 3rd-person boss-battle barrage

- Implemented animation system with VQS transformations and animation blending
- Wrote skinning shaders, draw methods, and post-processing for complex models (HLSL/DirectX)
- Designed and implemented FBX model conversion and asset pipeline tools from scratch using FBX SDK
- Co-architected and implemented engine (C++)
- Wrote majority of gameplay code (C++ and Squirrel)

Technical Skills

Languages - C/C++, C#, HLSL/GLSL, X86 (Reading)
Tools - Git, SVN, Mercurial, CVS, Perforce, Jira, Visual Studio, GCC/Makefiles, Command Line, Confluence
Mathematics - Multidimensional Linear Algebra, VQS Quaternions, Predicate Logic, Splines, Statistics
Familiar APIs - OpenGL, DirectX, FMOD(ex), FBX SDK, VRPN
Scripting - Python, AS3, Melscript (Pymel), JavaScript, Lua
Miscellaneous - Profiling, crash handling, technical documentation, networked simulation, optimization

Education

DigiPen Institute of Technology, B.S. CS and Real-Time Interactive Simulation **Graduated 2013**