Brian Ernst

the.bernst@gmail.com | http://leetNightshade.com | http://linkedin.com/in/leetnightshade

RELEASED TITLES:

PC: Star Citizen
NDS: World of Zoo, and Club Penguin: Herbert's Revenge
iOS: Oregon Trail American Settlers, Cosmic Colony, and Ice Age Village

COMPUTER SKILLS:

Languages: C++, C#, Python, Java, ActionScript 3, Lua
Engines: CryEngine, Gameloft's Glitch Engine (Irrlicht Engine based), UnrealEngine (limited), Unity 3D (limited)
APIs: DirectX 9/10 & HLSL, OpenGL 2/ES 1, Windows, WinForms, Windows/POSIX Threads, Winsock 2, iOS, Android 2.3

PROFESSIONAL EXPERIENCE:

Programmer @IllFonic *November 2013 – Present*

Star Citizen gameplay programmer. Handled player rotation and movement, including making CryEngine support zero-gravity player movement. Implemented magnetic boots (internally praised by Crytek) for walking up/down/around arbitrary surfaces. Initially hooked up all weapons and gadgets. Created grapple beam gadget with impulses to cancel extra velocity, speed player towards a wall, and slow them down before hitting it. Revamped weapon sway, extending it as player sway, from which I created procedural helmet sway to replace animation based helmet movement. More details to come.

Programmer @Gameloft *December 2011 – October 2013*

Worked on iOS games. Responsible for tutorial system, packaged/hooked in sound, extended data tool in ActionScript 3, performance improvements, dynamic loading/unloading sprites, fixed memory leaks, seamless log-in, hooked up social library that others couldn't, created a mini-game that generates Gameloft revenue, built fonts, built/uploaded DLC, managed a DLC release, did QA and Gold builds for App store. Headed up a licensed prototype project with Gameloft's 3D engine, hooked in all needed libs, maintained device builds, and set up an evolving Python-driven data pipeline that got faster as it developed to handle multiple times more data than Ice Age (over 2GB vs couple hundred MB), and it was still faster. Worked on Android game using Java with the C preprocessor: gameplay/GUI elements, took over archaic tutorial system, implemented cross game promo screen, hooked up tracking library, fixed z-order drawing, worked on Python scripts and C# tools that made once tedious work quickly manageable, and debugged build issues.

Programmer @1st Playable Productions *June 2009 – November 2009*

Used C++ to: create mini-games on World of Zoo; programmed a control scheme for Ben 10 that became an advanced unlockable; etc. Used C++ and Lua scripting on Club Penguin Herbert's Revenge: programmed the story sequences; set up items used for puzzles; added a scripting queue system; etc. Fixed bugs.

PROJECTS:

Creator of Warp, Cross-Platform C++ Engine *In progress*

Researching and reiterating on design of this platform and API independent multi-threaded engine. Area of interest is the Entity-Component model for main systems. Working on an efficient job system for easy scaling and asynchronous handling. Has components for DirectX 9/10 and OpenGL, which require tweaking before interface is finalized. More details to come.

Programmer for 2D Game Engine and Game Systems on PlasmaTech *Winter 2010 – 2011*

Set up rendering systems: abstracted Effect file, Sprite Sheet, Camera, Frustum, Mesh, and Mesh Object, etc. Wrote the Content Manager for loading textures and models. Added Math-helper libraries to go on top of the DirectX 10 libs (e.g. ray picking calculating screen-to-world space coordinates). Devised interface (used by Game Objects) from which components for physics, chasing, or other types of movement behavior evolved. Programming done in C++.

Engine programmer for Student Game Upwards *Summer and Fall 2010*

Conducted lots of research, looked at various open source engines. The engine is still alpha, but is cross-platform in nature, using components to implement API-specific functionality (see Warp engine, above). Also handled late GUI implementation. Programming done in C++, rendering using DirectX 9.

Programming for a 3rd Person Networked Shooter *Spring Quarter 2010*

Programmed in C# with XNA: Player Camera and Controls, setup Physics system with the Player, Network Manager and Messaging System, game statistics system that synchs over the network, flexible kill-feed that handles spamming text, and a test bed of the game environment for other programmers to work in.

EDUCATION:

Graduated from Rochester Institute of Technology (RIT), Rochester, NY, May 2011. Bachelor of Science in Video Game Design and Development. Cumulative GPA: 3.4 (4.0) Professional GPA: 3.6 (4.0)