

# Brian Ernst

the.bernst@gmail.com  
www.briansPort.com

## EDUCATION

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

## COMPUTER SKILLS

**Languages:** C++, C#, ActionScript 3, Java

## PUBLISHED TITLES

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

## PROFESSIONAL EXPERIENCE:

**Programmer at game development company, 1<sup>st</sup> Playable Productions** June 2009 - November 2009  
Used C++ to: create and program 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:

**Designer and Programmer for Warp, Cross-Platform Engine** Winter 2010 - Current  
Conduct lots of research to design and develop a platform and API independent engine. It uses Interfaces and Factories, where components are implemented as necessary. An active area of interest is the Entity-Component model for main systems. Animation components added: an Inverse Kinematics solution, key-framing, skinning, and soon blending. Threading and Networking Interfaces will be added, with Windows components for Winsocks2 and Windows Threading. Implemented components for DirectX 9 and 10, which require tweaking before the interface is finalized. Has a flexible, highly configurable control mapping system and conflicting keys system to queue keys meant for a single action; a combo key system is under research.

**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.

**Engine programmer for Student Game Upwards** Summer and Fall 2010  
Conducted lots of research, looked at various preexisting engines (e.g. Ogre 3D, Irrlicht, Allegro, and Wild Magic 3). The engine is still in alpha, but is cross-platform in nature, and uses components to implement API-specific functionality. To work well, certain classes/libs should be programmed as part of the engine, (i.e. the math libraries, messaging systems, thread manager, network manager, etc.) As this remains a massive endeavor, details can be provided upon request (Also see Warp engine, above).

**Programming for a 3<sup>rd</sup> Person Networked Shooter** Spring Quarter 2010  
Programmed: Player Camera and Controls, initial setup of Physics system with the Player, the Network Manager and Messaging System, game statistics system that synchs over the network, flexible kill-feed that handles spamming text well, and a test bed of the game environment for other programmers to work in.

**Programming for an RTS Game** Winter Quarter 2009-2010  
Programmed and researched path-finding, designed the spatial store and added features as needed for AI, implemented and added a basic flexible UI Form system for the GUI, implemented the flocking/boidal behavior, setup the units and basic AI, implemented resource mining, among other systems.