

Karl Gluck

817.938.1311 ▪ Austin, TX ▪ kwg8@cornell.edu ▪ karlgluck.com

Education

Cornell University College of Engineering, Ithaca, NY

Master of Engineering in Computer Science, May 2010. GPA 3.83

Bachelor of Science in Electrical and Computer Engineering, January 2010. GPA 3.74

Recognition Hunter R. Rawlings Cornell Presidential Research Scholar ▪ Dean's List ▪ Eta Kappa Nu (Hnr S)

Skills

Software	Visual Studio, Amazon S3, Linux, Apache, Git/hub, MySQL, Google Analytics, Photoshop
Languages	C++, JavaScript, Java, TCL, HTML 5/CSS 3, CSH, SQL, C, GLSL, PHP, SQL, x86 Assembly
Codebases	DirectX 9, Win32 API, jQuery/Mobile, Twitter Bootstrap, STL, Qt, OpenGL/ES, ENet, SQLite
Concepts	OOP, TDD, REST APIs, Scrum, 3D Rendering Pipeline, Bitcoin protocol, SW Optimization

Experience

Software Engineer, Intel Corporation

2012 – current

ATOM SOFTWARE PERFORMANCE, Austin, TX

- Streamlined analysis of 3000+ workloads by automating common tasks through a web interface and inventing a robust domain-specific programming language (DSL) used by team to process data
- Provided insight into a critical graphics performance issue which led to a product roadmap change
- Improved Android graphics benchmark performance 14% by optimizing OpenGL driver
- Developed internal website for managing cross-team power and perf. metrics (LAMP+JSON API)
- Created static code analysis tools that discovered 40+ race conditions in an Atom power MCU

Software Engineer, Intel Corporation

2010 – 2012

DESIGN TECHNOLOGY SOLUTIONS, Austin, TX

- Worked with team to maintain 128,000-line TCL codebase on front end of device-level CAD tool
- Developed robust automated regression system used to verify quality of each release
- Initiated use of Scrum techniques to improve collaboration and increase productivity
- Prevented a major chip logic redesign that would have pushed back the project's timeline
- Created a DSL used by 20 design engineers to hand-craft complex metal fill patterns

Programmer of Evidyon, a real-time 3D online multiplayer video game

2006 – 2010

SELF-INITIATED OPEN-SOURCE SOFTWARE (MIT License), <http://github.com/karlgluck/Evidyon>

- Authored 130,000 lines of C++ code in Microsoft Visual Studio, dedicating 3000 hours over 4 years
- Implemented client software with Win32 and DirectX APIs featuring GUI framework
- Wrote crash-tolerant multithreaded game server using ENet and SQLite
- Created a WYSIWYG editor to configure and preprocess more than 300 MB of media
- Developed an automatic FTP-based patching system and a remote server administration tool
- Incorporated feedback and managed an avid online community following
- Collaborated with a team to craft hundreds of items, create an Excel mechanics document using VBA, lay out a barter and auction economy, design AI-controlled inhabitants and moderate the game

Lead Software Developer, Fab@Home 3D Printer

2009 – 2010

CORNELL COMPUTATIONAL SYNTHESIS LAB, Ithaca, NY

- Designed an XML driver framework to allow an application to use more than one kind of 3d printer
- Led group meetings for a team of six programmers
- Assigned, discussed and reviewed programming tasks
- Delivered 12,000 lines of well-documented, modular, portable software using C++/Qt

Other Projects

See more on my Github page: github.com/karlgluck

Cryptasia

cryptasia.com

Cryptasia is a secure random password generator; it makes a unique password for every website using your secret phrase, so users never have to invent or remember passwords. It is unique because the end user hosts data in their own private Google Document, allowing every aspect of their passwords to be both customized and secured by the users themselves.

Player Upgrades

playerupgrades.com

Player Upgrades provides quizzes to help players of the online video game StarCraft 2 improve their strategy. It makes heavy use of jQuery and a customized version of Twitter Bootstrap.

TclTools

github.com/karlgluck/tcltools

This is a compact library with a lot of common time-saving functionality for Tcl/Tk applications. It features inline self-tests with Jasmine-like syntax, runtime assertions and several useful language extensions.

ThresholdJS

github.com/karlgluck/thresholdjs

ThresholdJS is a bare-bones implementation of Shamir's Secret Sharing algorithm for Bitcoin private keys. To make it easy to understand and verify, this implementation is just 140 lines including comments and whitespace.