

# Michael Vaganov (michael.vaganov@gmail.com)

Computer Programmer and Educator.

## Portfolio

<http://www.codegiraffe.com/portfolio> (diagrams & running code samples!)

## Software Development Skills

- Multi-platform: Unity3D, C/C++, Java, custom cross-platform frameworks
- Client side: User Interface, multi-skew builds, device debugging
- Server side: client/server serialization, database, web tools
- Rapid Prototyping: Unity3D, Command-line, PowerPoint, Excel, Pen+Paper+Dice
- Graphics: 3D (Blender), 2D (GIMP, MSPaint), procedural (assets from code)
- Simulations: 2D and 3D math, real-time systems (single / multi threaded)
- Data Structures: custom implementations for cache/memory/stability
- Scripting: JavaScript, LUA, SQL, custom languages + compilers + VMs
- Command-line: cmd in Windows, terminal in Linux/Unix, batch/bash
- Source Control: Git, SVN
- Web Development: Node.js, Heroku, Apache Tomcat, custom systems
- Build systems: batch, Apache Ant, make
- Mobile: Old Mobile (J2ME, BREW) and New Mobile (Android, Unity3D)

## Communication Skills

- English: public speaker, Polish: conversational, Mandarin Chinese: *yi dian dian*
- **10+ years as educator:** tutor and University Professor (taught age 7 to 40+)
- "Digital Native": started using computers at age 10, programming C++ at age 14

## Employment Time line

2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
Teaching @DeVry	pppppppppppppp	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	pppp`ppp-----						
Atlas/Infospace	FFFFF`-----	~~~~~-----	~~~~~-----	~~~~~-----	~~~~~-----	~~~~~-----	~~~~~-----	~~~~~-----	~~~~~-----
LimeLife	~~~~~	FFFFFFFFF`~~~~~	~~~~~-----	~~~~~-----	~~~~~-----	~~~~~-----	~~~~~-----	~~~~~-----	~~~~~-----
Other Teaching	~~~~~-----	~~~~~-----	~~~~~-----	~~~~~-----	~~~~~-----	ppp`pppp`-----	ppppppppppp`-----		
Impetus (stealth startup)	---	~~~~~-----	~~~~~-----	~~~~~-----	~~~~~-----	~~~~~-----	pppppFppppppppp`-----		

Key: (p) Part-time, (F) Full-time

## Employment Details

### Code Coach at theCoderSchool

September 2014 to present

Elite Computer Science education for youth (between age 7 and 17) in the Silicon Valley

### Self Employed Programmer, Entrepreneur

December 2012 to present

- Stealth Startup, working on project codenamed "Impetus"
  - Game + Project Management Software using Unity3D and C#
  - 3D model generation system for data visualization
  - Scripting system for data storage, content generation, AI
- Developed Node.js server backend (using Heroku) for Mehamagizmo's "Hangries"
- Developed OOP courses for DeVry's national GSP program
- Developed high-level curriculum strategy for DeVry's national GSP program

### Professor at DeVry University (Silicon Valley Campuses)

March 2006 to present

Professor of Games and Simulation Programming (GSP), a Computer-Science-like Bachelors of Science degree program, with emphasis on game development

- Rated highly in students evaluations (consistently 3.5+ out of 4)
- Managed 30+ Senior Project teams (16 week project, 2 to 5 programmers /team)
- Personal teaching style emphasizes:
  - Show them running code. Running Code Is Truth.
  - Leading by example (good code, honest testing, and honest communication)
  - Programming examples written in real-time, during class
  - Writing and testing working code from scratch, to show problem solving
  - Understanding the C/C++ memory model (basic Von Neumann Architecture)
  - High-level programming philosophy (see end of Resume)
  - Lab work as start-to-finish programming projects. They Must Write Code.

- Comparing academic and professional programming processes
- Agile and agile (small-a) development, favoring the small-a
- Joy-of-discovery and character-building-pain-of-debugging are important

Full list of DeVry courses taught at: <https://github.com/mvaganov/self/blob/master/RESUME.md>

## Software Engineer at LimeLife

*November 2006 to April 2008*

Developer responsible for end-to-end network-aware mobile application development, including Lead roles on build system, reusable framework API, game development tools, porting systems, and automation systems

- Simplified manual 4-step build process for each device to a fully automated build process for an arbitrary list of devices, using Apache Ant scripts and simple batch files to start processes
- Created automated OTA (Over The Air) deck generation scripts as part of J2ME build process, using PHP
- Created DRM (Digital Rights Management) abstraction layer for carrier/platform specific DRM systems, as well as client-side code for a custom, encrypted, carrier non-specific, DRM layer for LimeLife
- Acted as emergency porting engineer for "InStyle" and "Rachel Ray: Recipes on the Run" mobile apps
- Created a highly efficient 2D composite sprite format and renderer for mobile devices (both J2ME and BREW)
- Created Java-based GUI tool for creating and editing composite sprites
- Acted as technical artist for "Top Chef: the Mobile Game", building composite sprites and animations
- Part of senior development team that built ALE, a (quite impressive) wide-porting/localization/multi-platform (multi-lingual) API, and associated build systems
- Implemented garbage collection system used by C++ applications that were automatically ported from Java
- Created AML, an HTML-like scripting language used to describe UI and network-aware UI traversal for phones
- Created build tools, runtime engine (including container-based UI system), and on-the-fly server-side Java-based compiler for AML, a custom UI engine for mobile

Full list of LimeLife shipped titles at: <https://github.com/mvaganov/self/blob/master/RESUME.md>

## Software Engineer at Infospace Mobile Games

*December 2004 to November 2006*

Developer of mobile applications with emphasis on client/server interaction.

- Implemented and debugged multiple proprietary asynchronous Client/Server technologies.
- Trained engineers in proprietary BREW and J2ME technologies.
- Conceived and implemented original scriptable UI engines for mobile and created associated compilers.
- Developed zip-compression based networking/content distribution protocol.
- Designed, developed, maintained, and ported applications using "For Prizes" asynchronous multiplayer technology.
- "For Prizes" Expert - Acted as major knowledge store about proprietary For Prizes technology, including client/server transactions, and user registration and authentication processes, in both J2ME and BREW.
- Nominated for a company-wide Infostar award in the first 6 months of employment!

## Porting Engineer at Atlas Mobile (later purchased by Infospace)

*June 2004 to Dec 2004*

Very productive first 6 months of professional software development work

- Ported 5 "For Prizes" games to CDMA carriers and 30+ BREW devices.
- Prototyped a functional BREW UI engine.
- Identified as a 'BREW expert' by technical management, 6 months after learning BREW.

Full list of Infospace Mobile / Atlas Mobile shipped titles at: <https://github.com/mvaganov/self/blob/master/RESUME.md>

## CIS Tutor at DeVry University (Fremont Campus)

*March 2002 to October 2004*

CRLA certified, Tutor of the Semester (Summer 2003), Head Tutor (Fall 2004)

## Education

### Keller Graduate School of Management

*September 2006 to 2010*

Masters of Project Management

### DeVry University

*July 2001 to October 2004*

BS of Computer Information Systems

### Volunteering Teaching

- Citizen Schools, Joseph George Jr. High, (Feb. to Apr. 2014)
- Citizen Schools, Robert McNair Jr. High, (Oct. & Nov 2013)
- Coder Dojo Silicon Valley (Sep. 2013 to present)
- Guest Lecturer, Makerere University, Kampala, Uganda (Oct. & Nov. 2012)
- Guest Lecturer, Nkumba University, Entebbe, Uganda (Oct. & Nov. 2012)
- Guest Lecturer, Victoria University, Kampala, Uganda (November 2012)

- Introductory Game Programming with C/C++ (late 2011, 2012)

## Hobbies

- Rock climbing
- Volunteer Teaching
- Game Jams and Hackathons (sample works at [codegiraffe.com](http://codegiraffe.com))

## Personal Programming Axioms

- The best programmer writes the most Readable code. Speed is for the compiler.
- The best code will survive long after a programmer leaves it.
- Single Point of Truth: One complexity, One bug, One change.
- Code explicit functionality rather than side effects, and `/** document it */`
- Comments are good, code that describes itself is better.
- Think about optimization now, but do the actual optimization later.
- Just Prototype (and expect not to be given another shot at it).
- Program with a purpose: Understand the end-goal as soon as possible.
- Refactor sooner rather than later; clean code grows into powerful code.
- Disciplined, results oriented software development is always in style.
- How most production code should be judged (in order):
  - Functionality: intended results are produced (with constraints in mind)
  - Survivability: useable again elsewhere (maintainable/readable/modular)
  - Robustness: stability with a wide range of input (no bugs)
  - Resource Use: resources used conservatively (Big-O, memory, threads, ...)
  - Everything Else: elegance/robust-unit-tests/optimal-efficiency/...
- The Unix way feels right (<http://www.faqs.org/docs/artu/ch01s06.html>)

## Other Credo

- Persistence (iteration) is disproportionately important to success.
- Rules are for people who don't know any better; Rules are important, but understanding sets you free.
- Luck is where Preparation meets Opportunity.
- To make the next best thing, the current best thing must be mundane.
- A spoonful of test dissolves a pound of design.
- Do not fear complexity; simplify.
- <http://codegiraffe.com/quotes.txt>