Michael Vaganov (michael.vaganov@gmail.com)

Computer Programmer and Educator.

Portfolio

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

Skills

Software Development Skills

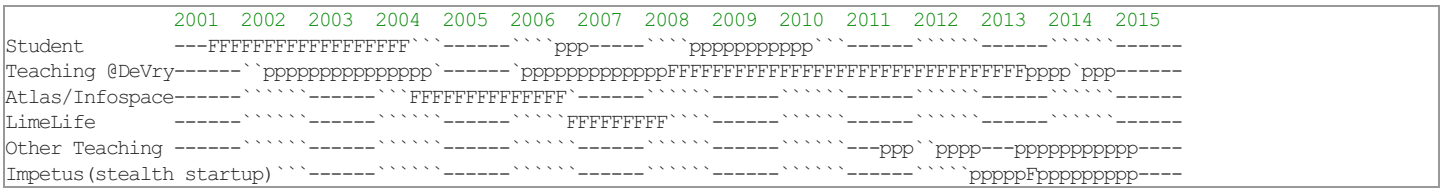
- Multi-platform: Unity3D, C/C++, Java, proprietary cross-platform frameworks
- Client side: User Interface, multi-skew builds, device debugging
- Server side: client/server serialization, database, web tools
- Rapid Prototyping: Unity3D, Command-line, PowerPoint, Excel, Pen+Paper+Dice
- Graphics: 3D (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, proprietary 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, proprietary 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": stated using computers at age 10, programming C++ at age 14

Employment History

Employment + Education Timeline



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

Empoyment Details

Code Coach at theCoderSchool

September 2014 to present

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

- Custom-built simple and engaging tutorial content for Computer Science and electronic art
- Subjects: Unity3D, C/C++, Java, Blender, 2D and 3D math, Project Management

Self Employed Programmer, Entrepreneur

December 2012 to present

Voluntarily left full-time job to pursue other ambitions

- 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 Mechamagizmo'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

- Faculty guide for game development clubs since March 2006
- Promoted from Adjunct to "Associate" Professor (full-time faculty) April 2008
- Voluntarily adjusted to "Visiting Professor" (part-time faculty) August 2013
- Extra-curricular game programming workshops for students
- Mostly on-campus classes, with experience teaching online classes as well
- Advised Dean while implementing new game development curriculum

DeVry Courses Taught:

- GSP110/GSP111 - Introduction to the game development process
    - Overview of games industry history & culture
    - Practical game and level design
    - Game Development Life Cycle
    - First semester course: freshman mentoring
- CIS115/GSP115 - Early programming course taught in C/C++
    - Introduce programming fundamentals, in the context of games
    - Emphasis on practice. Early memorization comes naturally with practice.
- GSP125 - Intermediary and Object Oriented Programming
    - OOP, including composition, inheritance, and polymorphism
    - Pointers and the C/C++ memory model
    - Developing simple games using OOP techniques
- GSP240 - Game Design
    - Game analysis, mechanics, development, Game Design Documentation
- GSP261 - Computer Graphics and media
    - 3D modeling with Blender
    - Character rigging, particularly bipedal models
    - Texturing, materials, shader basics, lighting
    - Simple sound and music development for games
- GSP280 - Simulation and Design with Lab
    - Basics of practical computer simulation development
    - Heavy lab focus (producing lots of working code from scratch)
    - 2D Collision (circular, polygonal, convex, bullet-through-paper)
    - SDL (Simple Direct-media Layer) abstraction layer
- GSP290/GSP295 - Data Structures (and AI)
    - Data Structures (Linked Lists, Vectors, Hash-tables, Trees, Graphs, ...)
    - Applied Von Neumann Architecture as the C/C++ memory model
- GSP315 Artificial Intelligence
    - Artificial Intelligence (Steering Behavior, A* path finding, ...)
    - Scripting, Behavior trees
    - OpenGL vector graphics
- GSP340 - Level Design
    - Game Level creation (including scripting) using 2D and 3D game editors
    - 2D tile-based game engine (from scratch), using Notepad as an editor
- GSP360/GSP361/GSP362 - Mid-term/Applied Project
    - Start-to-finish Game development with inexperienced student teams
    - Applied Game Development Life Cycle, Scrum, agile
    - Source control use and collaborative-programming practices
- GSP420 - Game Engine Architecture and Design course
    - Game software components (Memory management, Graphics, Scripting, ...)
    - Practical game implementation from scratch, in C/C++
    - Design patterns and architecture in Java
    - Code Optimization in C/C++
    - Software Development Philosophy
- GSP490/GSP494/GSP497 - Senior Project Course

    - Final course in the GSP curriculum
    - Practical Applied Game Development Life Cycle
    - Applied Software Development Processes (Iterative/Agile)
    - Management of multiple student-driven game development teams

- Online Teaching Experience

    - Equivalents of on-campus courses: GSP115, GSP125, GSP295, GSP360, GSP490
    - Threaded discussions, video conferencing, and YouTube
    - Mandated course content, using eCollege, supplemented with custom content.

- GSP1337 - Extra-curricular volunteer game development courses

    - Un-scheduled, done during vacation, content in response to student needs
    - Usually focusing on networked game programming in C/C++ with WinSock
    - Often data structures review (Graphs in particular)

- Game Programming with Unity3D and Blender (outreach class)

    - Extra-curricular High-school level course
    - Tutorial: making simple games with Unity3D, C#, and Blender

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

LimeLife Mobile Titles List:

- Hallmark Smiles + Styles - J2ME/BREW - Lead Developer - Web-aware browser that used scripting engine to display network distributed UI. Built and designed from the ground up in both J2ME and BREW, including multi-platform HTML-like scripting language and compiler, container-based UI system, and platform abstraction code (including multi-lingual network protocol abstraction), which was integrated into main LimeLife porting framework (ALE).
- Urban Chica - J2ME/BREW - Lead Developer - A re-skinned application built in parallel and accomplished within *hours* of final builds of Hallmark Smiles + Styles. No source code alterations needed, only script modifications.
- Top Chef, the Mobile Game - J2ME/BREW - Framework Engineer - Designed and built composite Sprite engine. Also acted as Animator and technical artist.
- Rachel Ray: Recipes on the Run - J2ME/BREW - Porting Engineer - Network aware application used to find and share licensed Rachel Ray recipes. Utilized FLIRT, a LimeLife proprietary scripting and UI layout technology
- InStyle - BREW/J2ME - Porting Engineer - Network aware application used to distribute InStyle magazine content, based on FLIRT, the same technology used to build Rachel Ray.

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.

Porting Experience With The Following Phones (not all phones listed):

- BREW - Audiovox (CDM8910, CDM8940, CDM8600, CDM8900), Kyocera (KX1, KX2, KX444, SE47), LG (VX4400, VX4500, VX4600, VX4700, VX6000, VX6100, VX7000, VX8000, VX8100, VX8500, VX10000), Motorola (V65, V260, V265, T720, C343, V710, V262, E815, V3, K1), Samsung (N330, A610, A790,

A650, U740).
- J2ME - MIDP-1.0 and MIDP-2.0 (Sony Ericssons, LG, Samsung, Motorola V series, Nokias, ...)

Infospace Mobile / Atlas Mobile Titles List:

- Tetris Tournament For Prizes - BREW - Porting Engineer - Tetris with a "For Prizes" component. One build worked on every handset tested. This portability was implemented above and beyond spec and enumerated work items (drawing and UI resized, had multiple handset specific bugfixes that were benign on all other devices).
- Prize21 For Prizes - BREW - Porting Engineer - Fast paced puzzle game. Experimented with procedural drawing with some great results. One build worked on every handset tested (as Tetris Tournament).
- QBz for Prizes - BREW - Porting Engineer - Port of a popular web game. Experimented with BREW framework design. Refactored art system and data structures for better scalability.
- Holdem Poker Plus For Prizes - BREW - Porting Engineer - Puzzle game with Hold 'em Poker theme. Heavy art and layout refactoring for small and large phones.
- Boulderdash - BREW - Support for Porting Engineer - BREW port of popular old-school PC game. Trained a porting engineer by supporting his port of this game.
- AMF Bowling For Prizes - BREW - Support for third party developer - Offered support to out-of-house developers using the For Prizes SDK
- Trickshot Pool For Prizes - J2ME - Framework Engineer - Pool game where the player is required to make trick shots. Wrote For Prizes module based on older J2ME For Prizes framework. Optimized away 10kb of code from a 30kb J2ME "For Prizes" framework (compressing J2ME API byte-source by 33%, post obfuscation, post jar compression).
- Skeeball For Prizes - J2ME - Framework Engineer - Provided support for J2ME For Prizes framework (same one used in Trickshot).
- Hotties - J2ME - Lead Developer - Massively Multiplayer phone blogging + Hot or Not game. Developed, from scratch, a scriptable UI and logic engine for J2ME (including a new scripting language and script compiler) as a "shippable prototype". Optimized and increased the functionality of a proprietary client/server protocol. Wrote Java Bean server components that interacted with the client on Tomcat servlets. Trained 2 engineers on how to use the scripting language (one server side, and one client side). Created server logic and compiler tools to automatically generate script files to be streamed to handsets with dynamic content and UI.
- Survivor Island Mobile - J2ME - Developer - Wrote a J2ME network-aware resource-loading API and zip-decoder, including resource manager to enable the phone client to grab compressed game resources from the web, using HTTP, and cache data (based on phone RMS capability). "For Prizes" Integration. Optimized and debugged.
- TestT4P - J2ME - Lead Developer - Robust J2ME "For Prizes" testing application. J2ME application's binary had near universal portability: from the Nokia 6010 (tiny screen, only 2 direction buttons), to the top-of-the-line (of the day) Samsungs and Sony Ericsons. Written as an instructional tool to teach new-hire developers and QA team how to test and port "For Prizes" games, and how to write J2ME code (built with straight forward design and good J2ME design principles). Became central testing point for a programming test to upgraded QA to porting engineers.

CIS Tutor at DeVry University (Fremont Campus)

March 2002 to October 2004 CRLA certified, Tutor of the Semester (Summer 2003), Head Tutor (Fall 2004)

- Tutored hundreds of students, primarily in algebra, and computer science
- Trained and mentored new tutors, and wrote training documentation

Education

Keller Graduate School of Management

September 2006 to 2010

Masters of Project Management

- Part time classes
- Practice with business and management concepts including:
    - Work Breakdown Structures, Gantt, Decision trees, RACI
    - Risk Management, Quantitative Decision Making, 6 Sigma
    - Agile, Marketing Analysis, Budgeting, Leadership

DeVry University

July 2001 to October 2004

BS of Computer Information Systems

- Graduated GPA 3.76 (Deans List)
- Special Honors
    - Summa Cum Laude
    - Awarded "Excellence in English and Humanities"
    - Graduated Tutor (2003 Tutor of the Semester)
- Team Leader and Lead Engineer for award winning senior project: "Nizzols"
    - Real-time Java Applet multiplayer game using TCP/IP sockets
    - Awarded "Complexity of Design"
- DeVry Advocate (volunteer program organized by student services)
- Clubs: Chess Club member, Game Development Club President
- Staff at ACM programming competition hosted at DeVry

Other

Volunteering Teaching

- Citizen Schools, Robert McNair Jr. High, (Oct. 2013)

    - Assisted a "Global Citizens" class, where students research world issues

- Coder Dojo Silicon Valley (Sep. 2013 to present)
    - Teaching Computer Science to kids 10+
    - Scratch, Python, Web Development, Unity3D

- Guest Lecturer, Makerere University, Kampala, Uganda (Oct. & Nov. 2012)
    - Professional Game Programming with C++
    - about 250 students (under-graduate and graduate)

- Guest Lecturer, Nkumba University, Entebbe, Uganda (Oct. & Nov. 2012)
    - Game Programming with Unity3D
    - 30+ students (high-school and under-graduate)

- Guest Lecturer, Victoria University, Kampala, Uganda (November 2012)
    - Introductory C++
    - about 20 students (under-graduate)

- Introductory Game Programming with C/C++ (late 2011, 2012)
    - Included a Unity3D tutorial
    - Neighborhood volunteer: 4 students (middle-school and adult), in late 2011
    - River of Life Church: about 20 students (middle-school) in mid-to-late 2012

Hobbies

- Rock climbing
- Volunteer Teaching
- Game Jams and Hackathons (sample works at `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`