Michael Vaganov (michael.vaganov@gmail.com)

Summary

Senior level software developer with game programming, teaching and technical mentoring background. 8+ years of teaching undergraduate game development in C++, in technical and practical 8-week courses, and guiding final projects (rapid prototyping with jr developers). 4+ years teaching CS K-12, primarily High School. 6+ years Teaching Unity 3D game/VR development. 2+ years technical management. Professionally implemented Unity prototypes, garbage-collection systems in C++, socket networking, domain specific scripting languages, and a custom web-browser on pre-smart-phone mobile.

Skills

- **20+ years Programming**: hobbyist, game programmer, educator, consultant, software engineer
- Programming Languages: **C, C++, C#, Java, JavaScript, Python**
- Software Domains: games, productivity, automation, UI/UX, client/server, 3D, AR/VR/XR
- Rapid Prototyping: Unity, JavaScript, Google Docs, Pen+Paper+Dice
- Graphics: 3D modelling (Blender), 2D (Photoshop/GIMP), procedural (assets from code)
- Simulations: 2D and 3D math, real-time systems (single & multi threaded)
- Data Structures: custom implementations for cache/memory/fragmentation
- Scripting: LUA, SQL, custom parsers + languages + compilers + VMs
- Dev tools: Command-line (cmd, terminal, batch/bash), Git
- Web: HTML5 + CSS + JavaScript, Angular

Teaching

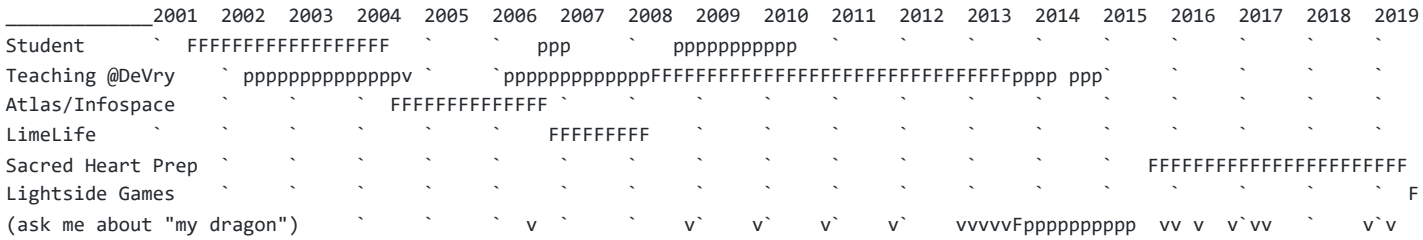
- **20 years Teaching Computer Science**: ages 7 to 40+, as tutor, undergrad professor, high-school teacher, code coach
- 4 years teaching High School Computer Science and tech oriented project based learning
- 8 years teaching Undergraduate Computer Science (Game and Simulation Programming)
- 7+ years tutoring/mentoring, at The Coder School, at Applied Computing Foundation, and college.

Portfolio

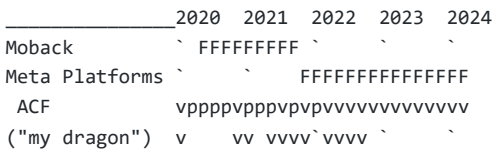
- Code Samples: https://github.com/mvaganov/
- Personal Website: http://www.codegiraffe.com/portfolio/

Employment History

Pre-COVID Employment + Education Timeline



Contemporary Employment Timeline



Key: (F) Full-time (40+ hrs/wk), (p) Part-time (~20 hrs/wk), (v) Volunteer (~10 hrs/wk)

Employment Details

Software Engineer at Meta

(*Nov. 2021 to present*) Working on tech infrastructure for the metaverse

- Originally hired as a Contingent Worker via Crystal Equation, converted to Full Time Employee June 2022
- Maintained prototyping infrastructure (Unity packages) for prototypers, including testing and feature development
- Created user interface tools in virtual reality for on-device debugging of networking systems
- Designed and implemented prototypes testing XR APIs and device features (realtime multiplayer, face/eye tracking)
- Implemented public SDK samples for Meta's (XR) Movement SDK
- Familiar with Meta engineering infrastructure (the mono repo and diffs, security and provisioning, VR/AR hardware)

Senior Software Engineer at Moback

(*Apr. 2020 to Nov. 2021*) Member of a mercenary guild of elite robotics/VR/AR developers

- Developed Augmented Reality (AR) software prototypes using Unity, C#, C, C++, and Python
- Created compelling illusions that exist in Oculus VR, and custom multimedia devices including LED arrays, speakers, and haptics
- Developed realtime simulation serving data to hardware and software clients, including Maya and Blender, using sockets
- Wrote many iterations of Teensy micro controller app for novel hardware, including memory management & realtime communication
- Technical Project Management: defined tasks, mentored engineers, found key hires, vetted new hires in technical interviews
- Collaborated with fortune 500 company client to cast vision and design science-fiction-like augmented reality tools

Director of Engineering at Lightside Games

(*Jun. 2019 to Apr. 2020*) Ranking software developer at a distributed mobile games studio

- Developed proof-of-concept software prototypes with Unity
- Planned engineering culture and advancement track for engineers, including review process
- Vetted external contractors responsible for bulk game development
- Collaborated with lead designer to plan product pipeline, including concept market-testing
- Represented company at conferences and industry meetups

Chief Educator, Futurist, and Code Coach at Applied Computing Foundation

(*Feb. 2020 to present*) Showing how to Apply Computing to solve problems (I'm making wizards!)

- Part of executive committee, giving feedback on process and guidance on future initiatives
- Teaching Virtual Reality, Game Development, and Entrepreneurship classes

Faculty at Gamebridge Unityversity

(*Aug. 2016 to Mar. 2020*) Lead weekly Virtual Reality & Game Development workshops

- Responsible for Unityversity's Santa Clara Central Park Library classes in Santa Clara, CA (2019)
- Improvisational tutorials about: programming, game design, software development, math, 3D modelling, digital art, other tech.
- Taught throughout the California Bay Area (USA), and Seoul (South Korea, Nov 2019)

Computer Science Teacher at Sacred Heart Prep

(*Aug. 2015 to Jun. 2019*) Faculty member at an exclusive private school.

- Taught computer science, with curriculum designed to motivate with creativity and computer hacking
- Subjects: Computer Science fundamentals, HTML/CSS, Python, JavaScript, C, C++, C#, Unity, VR, computer graphics, photoshop, 3D modeling, 3D printing, electronics, cyber security, tech industry culture
- Wrote software used to notify complex class schedule using speech synthesis
- Wrote Artificial Neural Network from scratch using Python, then again C#, as a learning exercise

Sacred Heart Courses Taught:

- Exploring Computer Science - introductory computers course
  - Series of practical exercises and presentations to learn about modern computer technology
  - Graphics, Web Development, Cyber Security, Programming (Scratch, Python), technology trends
- Programming - Python lab and algorithms course
- Building Virtual Worlds - Unity 3D and Oculus Rift, create & present Virtual Reality experiences
- Creative Inquiry - student-directed extra-curricular study, supported and graded by faculty

Code Coach at theCoderSchool

(*Sep. 2014 to Aug. 2015*) Elite Computer Science education for youth (between age 7 and 17) in the Silicon Valley.

- Custom-built tutorials for: Unity, C#, C and C++, Java, Blender, 2D/3D math, Game Design, Project Management

Self Employed Programmer, Entrepreneur

(*Dec. 2012 to Present*) Personal moonshot, and consulting services including prototyping and technical planning.

- Stealth startup project: Game & Project Management Software (unfinished)
  - 3D model generation system for data visualization
  - Scripting system for data storage, content generation, and expert-system AI
- Contracted consulting work
  - 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
  - Vetted engineering hiring process and technical management at Chefee

Professor at DeVry University (Silicon Valley Campuses)

(*Mar. 2006 to Dec. 2014*) 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), despite having difficult classes
- Focused on teaching performant code and game development in C and C++
- Taught and evolved course content: programming, data structures, practical software architecture, computer graphics, AI for games (expert systems), project management, design
- Managed 30+ Senior Project teams (16 week projects, 2 to 5 programmers /team with varying skill levels)
- Personal teaching style emphasizes:
  - Understanding is the price that must be paid. 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 and mentor for game development clubs
- Promoted from Adjunct to "Associate" Professor (full-time faculty) Apr. 2008
- Voluntarily adjusted to "Visiting Professor" (part-time faculty) Aug. 2013
- Extra-curricular game programming workshops for students
- Taught on-campus classes in Fremont and San Jose, and online classes
- Advised local and national Deans while implementing new curricula

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 and 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 lectures
    - 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

(*Nov. 2006 to Apr. 2008*) Developer responsible for end-to-end network-aware flip-phone mobile application development.

- Senior-level engineer: product development, build-systems and automation, client/server
- Fully automated heavily manual build process requiring test-activity to seed meta-data, saving hours-per-day for build engineers
- Created DRM (Digital Rights Management) abstraction layer, implementing client and server side code
- Implemented garbage-collection and a domain-specific web-browser for "ALE", a (quite impressive) wide-porting/localization/multi-platform, multi-lingual (C++ and J2ME) API and build system. Created for flip-phones, ALE's build system could notably compile a J2ME program into BREW C++ for any target device known by the system.
- A responsible part of shipping 5 distinct mobile titles, and many SKUs of each
- Created AML, an HTML-like scripting language used to describe UI and streamed-UI traversal for phones
- Created automated OTA (Over The Air) deck generation scripts as part of J2ME build process, using PHP
- Acted as emergency porting engineer for "InStyle" and "Rachael 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

- 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.
- Rachael Ray: Recipes on the Run - J2ME/BREW - Porting Engineer - Network aware application used to find and share licensed Rachael 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 Rachael Ray.

Software Engineer at Infospace Mobile Games

(*Dec. 2004 to Nov. 2006*) Developer of mobile applications with emphasis on client/server interaction.

- Senior-level engineer: product development, framework, R&D, client/server
- Implemented and debugged multiple proprietary asynchronous client/server products
- Conceived and implemented original scriptable UI engines for mobile and created associated compilers and virtual machines
- Lead development of a social-media photo-blogging application
- Created a client/server test app used for system testing and engineer training
- A responsible part of shipping 5 distinct mobile applications, and many SKUs of each
- Trained engineers in proprietary BREW and J2ME technologies
- 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)

(*Jun. 2004 to Dec 2004*) Very productive first-6-months-of-professional-software-development.

- Client side QA developer, primarily tasked with porting and bug fixing
- Identified as a 'BREW expert' by technical management, 6 months after learning BREW
- A responsible part of shipping 5 "For Prizes" mobile titles, and 30+ SKUs of each
- Prototyped a functional BREW UI engine

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

- BREW - Audiovox (CDM 8910, CDM 8940, CDM 8600, CDM 8900), 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 compiled binary 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 bug fixes 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.

Education

CIS Tutor at DeVry University (Fremont Campus)

(*Mar. 2002 to Oct. 2004*) CRLA certified, Tutor of the Semester (Summer 2003), Head Tutor (Fall 2004)

- Tutored 100+ students, primarily in algebra, and computer science
- Trained and mentored new tutors, wrote training documentation

Keller Graduate School of Management

(*Sep. 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

(*Jul. 2001 to Oct. 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

- Computer Science Teachers Association, Silicon Valley CSTA chapter (Member: Jan. 2017 to 2020, President: Jan. 2018 to Sep. 2019)
- Gunn Highschool Computer Science Capstone (Jan. & Feb. 2021)
  - Taught C# command line game programming and Unity to about 30 students
  - co-teacher with Josh Paley
- FIRST Robotics, FTA (Field Technical Advisor) certified (Jan. 2020)
  - Atherton CA Qualifier
- Citizen Schools, Joseph George Jr. High (Feb. to Apr. 2014)
  - Co-lead "Computer Animation", students learned programming with Scratch
- Citizen Schools, Robert McNair Jr. High, (Oct. & Nov. 2013)
  - Assisted a "Global Citizens" class, where students research world issues
- Coder Dojo Silicon Valley (Sep. 2013 to 2017)
  - 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 (undergraduate 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 (Nov. 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

- Hiking, Biking, Rock Climbing, Fencing
- Software Side-projects, Game Jams and Hackathons

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 don't expect another shot at it, so make it good!
- 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. (So, iterate. Faster.)
- Rules are for people who don't know any better; Rules are important, but Understanding sets you free.
- Luck is where preparation meets random opportunity, which is happening constantly.
- To make the next best thing, the current best thing must be mundane.
- A spoonful of test dissolves a pound of design.
- Without clear goals we are wasting people's time, and we are made of time.
- Do not fear complexity, simplify it.
- more at: http://codegiraffe.com/quotes.txt