RESUME

Jukka Jylänki

In short:

- Born in 1985.
- Software Engineer with a strong focus on C++, graphics programming, Computer Science and Mathematics.
- Holds a Master's Degree in Mathematics from the University of Oulu, Finland.
- Currently lives in Oulu.
- Webpage at http://clb.demon.fi/

jujjyl@gmail.com

Work Experience

2008-Present Senior Programmer and a Team Lead At LudoCraft Ltd.

- Technology Lead on the RealXtend Tundra project. Designed and validated the overall architecture and APIs.
- Technology Lead on the Project Room virtual meeting space. Implemented voip and video streaming, and webcam and screencast sharing.
- Special effects supervisor on the Miivies project. Developed character systems, shaders and special effects.
- Lead programmer on the GameBridge and Sandbox educational game projects. Wrote engine and gameplay code.

2005-2008 Senior Programmer, game engine developer and AI developer at Farmind Ltd.

- **Puzzle Scape, PSP:** Ported and re-developed a Direct3D9-based graphics and game engine for use on the PSP platform.
- **World Series of Poker 2008, Nintendo DS:** Co-developed a game engine for the NDS platform. Implemented the AI opponents.

Education

2004 Graduated in Finnish upper secondary school, Oulaisten Lukio, with a GPA of 9.7/10, and four grades of Laudatur in A-level Mathematics, English, Swedish and Physics/Chemistry. Received a scholarship prize for excellence in natural sciences.

2004 Served nine months in Rovaniemi LapItr as Military Police, attaining the rank of Corporal.

2005-2007 Studied at the Department of Information Processing Science at the University of Oulu.

2008 Studied a semester of Computer Science and Mathematics at the University of Newcastle in Australia.

2011 Graduated B.Sc. and M.Sc. from the Department of Mathematics at the University of Oulu, with a grade of 5/5. Received a scholarship prize from the Tauno Tönning Foundation for providing novel research in the Master's Thesis "*Check Character Systems over Algebraic Groups*".

Technical Skills

Highly skilled with: C/C++, Direct3D9/10/11, GLES2, OpenGL 3.1, Cg/GLSL/HLSL, TCP/UDP, Qt, Ogre3D, Visual Studio, SVN, Git, 3D mathematics, algorithms and data structures, parallel programming, Android NDK.

Experienced with: C#, SSE/SIMD, Boost, HTML/XML/JSON/CSS, JavaScript, Java, .Net Framework, Unity 3D, Mercurial, PHP, MySQL.

Some experience with: Assembly, Python, WebSockets, WebGL, ARM NEON, XCode and OSX+iOS programming.

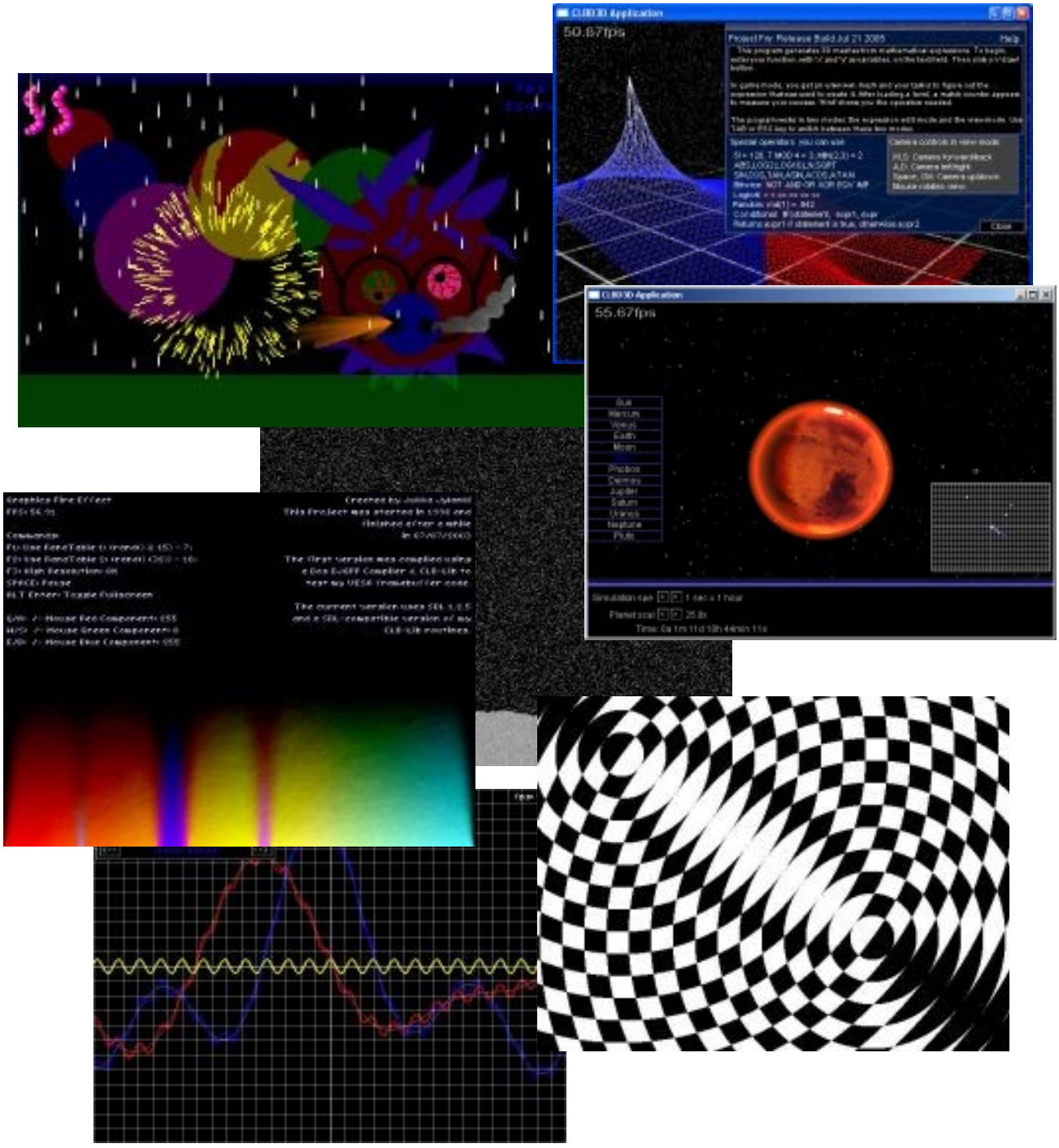
Achievements

- The author of **MathGeoLib**, an open source C++ library for 3D matrix-vector algebra and geometry manipulation. Available under the Apache 2 license from github repository juj/MathGeoLib. Runs on Windows, Linux, Mac and Android.
- The creator of **kNet**, an open source C++ networking library for games and real-time streaming applications. Runs on Windows, Linux and Mac. Implements reliable UDP, message prioritization and multichannel messaging. Available at Bitbucket in the repository clb/kNet under the Apache 2 license.
- Self-published research *A Thousand Ways to Pack the Bin - A Practical Approach to Two-Dimensional Rectangle Bin Packing*.
- Master's thesis available at http://clb.demon.fi/Gradu_Jylanki_final.pdf (in Finnish).

PORTFOLIO

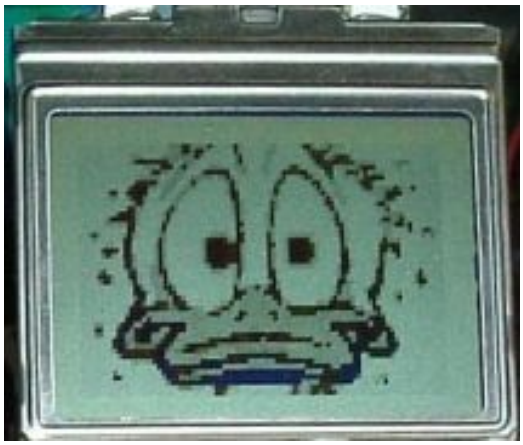
Jukka Jylänki

When I don't write code at work, I most often write code at home. This is a sampling of the different projects I have developed.



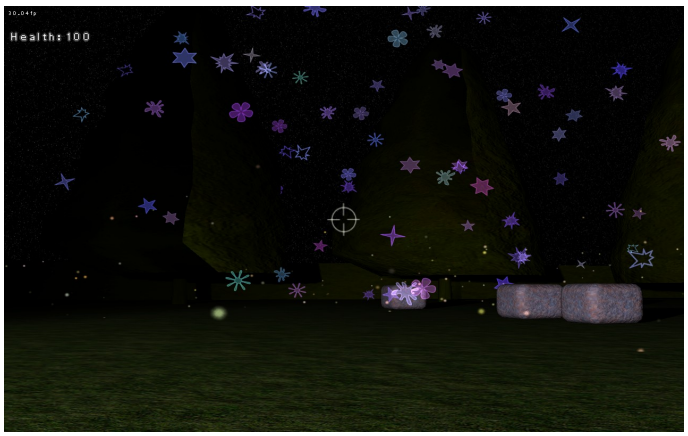
Old pictures of projects from the DOS era: Different framebuffer-based demo effects, a worm game, a 3D function graph plotter, and a planetary simulator. I started programming somewhere

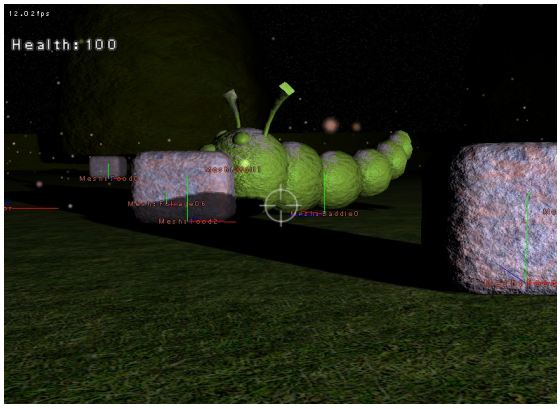
around 1997.



I can understand spec sheets for low-level hardware devices, and write programs to utilize them. I have also programmed Polar heart rate monitors and NMEA-based GPS devices over a Bluetooth serial port data link.

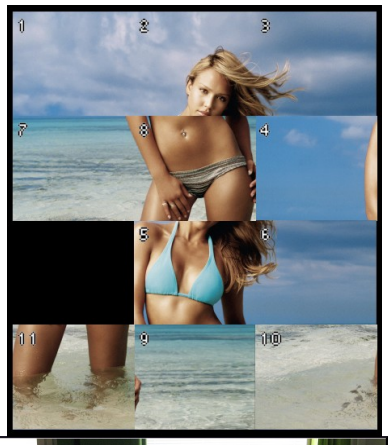
I have a strong grasp of the Minimax algorithm, along with Alpha-Beta heuristics, iterative deepening and transposition tables, and can develop and maintain an implementation that performs game tree searches. In the picture is a Direct3D 7-based chess game I developed in 2004, which includes a Chess AI that explores about 500K game nodes per second on a single core. I am also familiar with Monte Carlo techniques in the context of Computer Go.





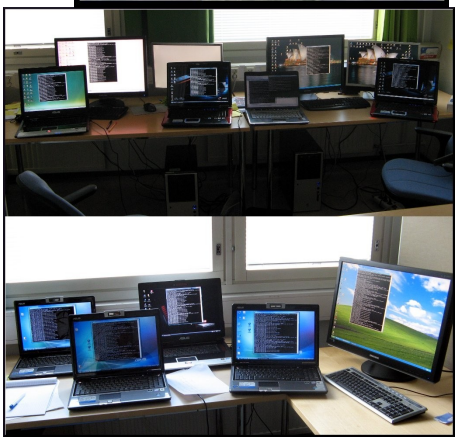
In 2007, I wrote a demo game for an University course to showcase my game engine then-in-development. It included a Direct3D9 renderer with particle systems, normal mapping, shadow

mapping and bitmap-based font rendering.



I am familiar with the implementation details of various computational algorithms. I understand the Big-O notation, and can manipulate it with formal rigor. I have experience with implementing data structures e.g. for priority queues and search trees, and have no problems with running mathematical proofs with induction, or understanding the difference between normal and tail-free recursion.

I am comfortable with implementing computer search methods, such as the A* algorithm for pathfinding. The image on the side shows the A* search applied to solve the sliding puzzle.



I am comfortable with parallel programming, and am familiar with parallel constructs such as mutexes, semaphores, critical sections, as well as low-level synchronization primitives such as atomic increments and compare-and-exchange operations. I understand the concepts of wait-free and lock-free.

In 2009, I took part in one of “Al Zimmermann's Programming Contents” on combinatorial optimization, and implemented a massively distributed software to solve a hard problem on discrete integer optimization.

Point Packing

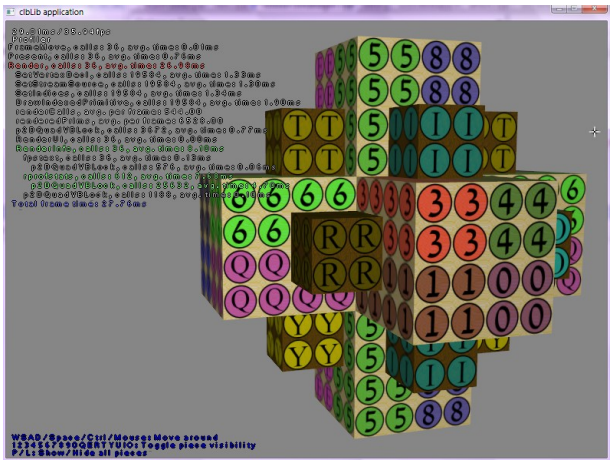
Ends: 10 Oct 2009 16:00
Now: 24 Jul 2009 22:14

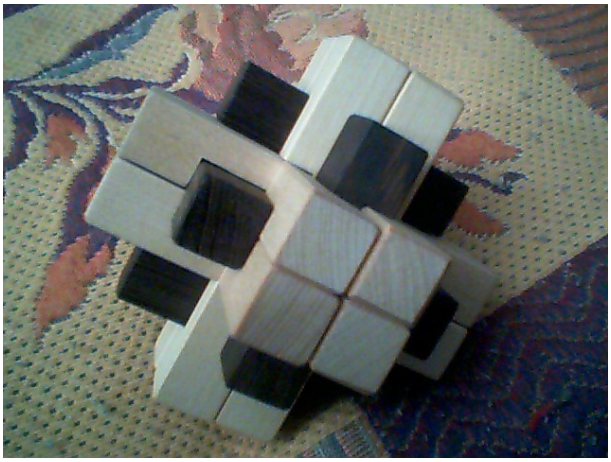
Description | Submit an Entry | Standings

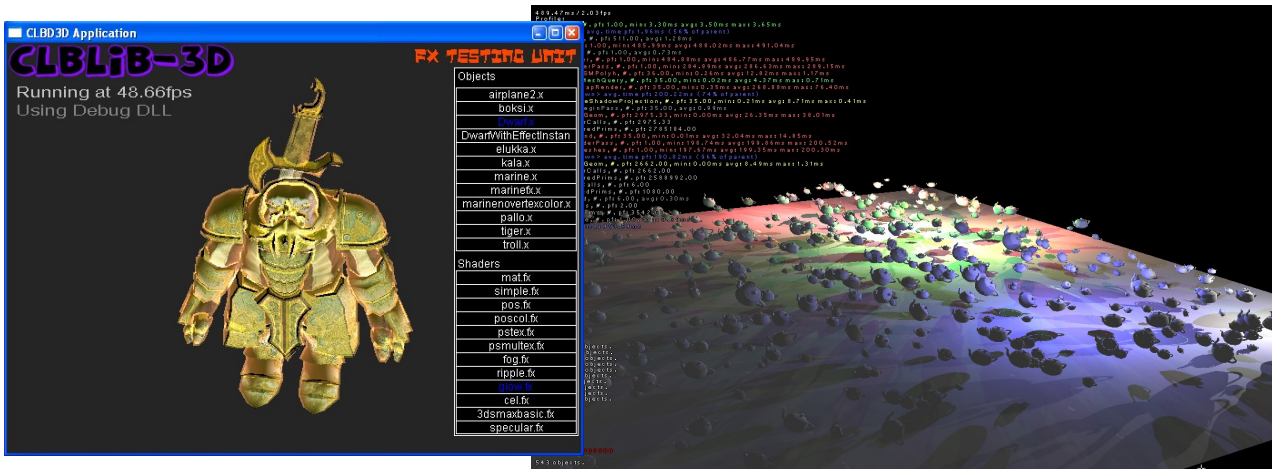
Rank	Score	Contestant		Last Improvement
1	24.98	**Jukka Jylänki**	Oulu, Oulu, Finland	24 Jul 2009 22:05
2	24.66	**Garr Godfrey**	Seattle, Washington, United States	24 Jul 2009 00:59
3	24.30	**Ivan Kazmenko**	Saint-Petersburg, Russia	24 Jul 2009 12:38
4	24.07	**Jim Gillogly**	Maui, Hawaii, United States	24 Jul 2009 18:41
5	24.06	**Wes Sampson**	La Jolla, California, United States	24 Jul 2009 05:59
6	23.94	**James Dow Allen**	Uthai Thani, Thailand	22 Jul 2009 05:42
7	23.91	**Wladimir Leite**	Sao Paulo, Brazil	22 Jul 2009 12:40
8	23.85	**Mark Mammel**	Ellicott City, Maryland, United States	24 Jul 2009 20:39
9	23.57	**Giuliano Bertoletti**	Parma, Italy	24 Jul 2009 20:07
10	23.53	**Rémi Coulom**	Lille, France	21 Jul 2009 13:40
11	23.49	**Patrick Domichel**	Grenoble, France	22 Jul 2009 20:12

I led the contest for three months, but eventually finished in the fourth place. At the best time, I had over 300 computing cores distributed over the internet searching over the solution space.

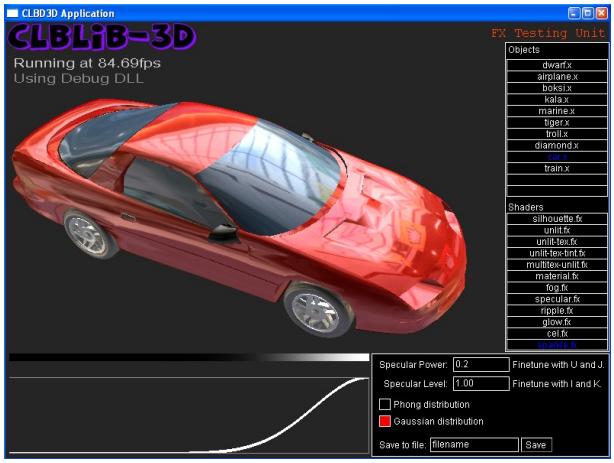
I took apart my friend's 18 piece burr puzzle (pirunnyrkki), and had to write a computer program with a backtracking solver to put it back together, since my friend had no knowledge of how to assemble it.

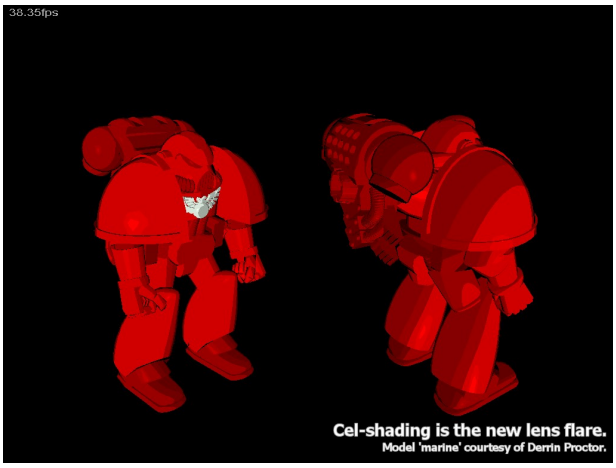


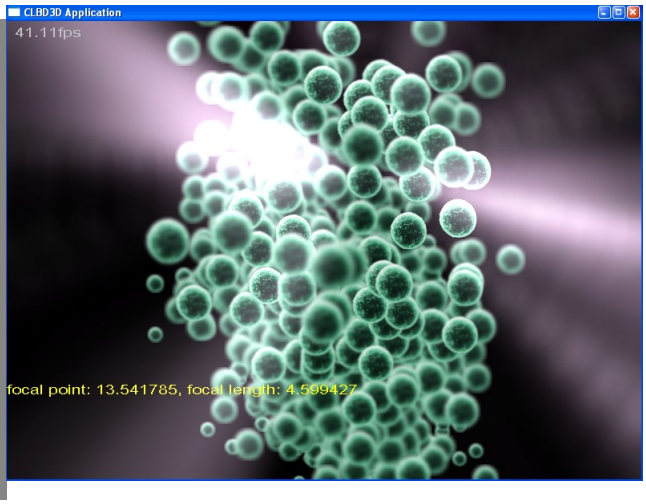


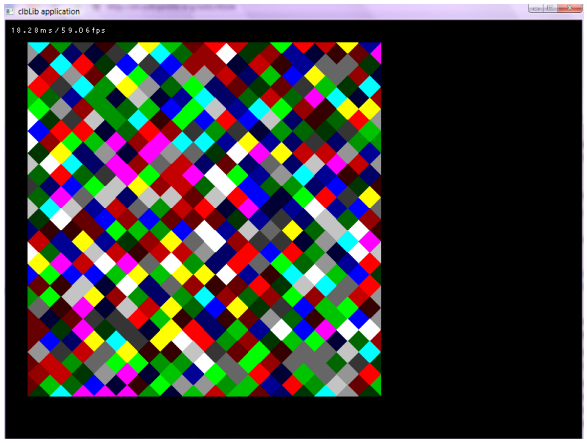


I am familiar with GPUs, and I understand how to implement lighting equations in shaders, with effects like specular lighting, reflection mapping, tangent-space light calculations, normal mapping and cel-shading (toon rendering). I can implement rendering pipelines for framebuffer effects like glow, bloom or HDR.



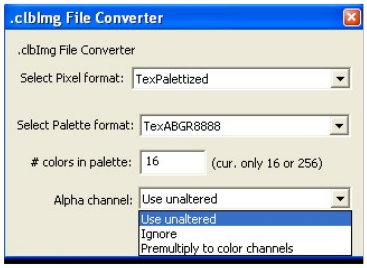


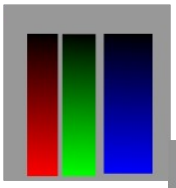


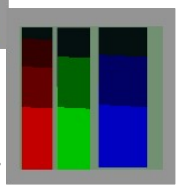


I have written a solver for [the Eternity 2 puzzle](#) (and can safely conclude that it is impossible)..

… implemented Median Cut (a true-color to palettized image quantizer)...



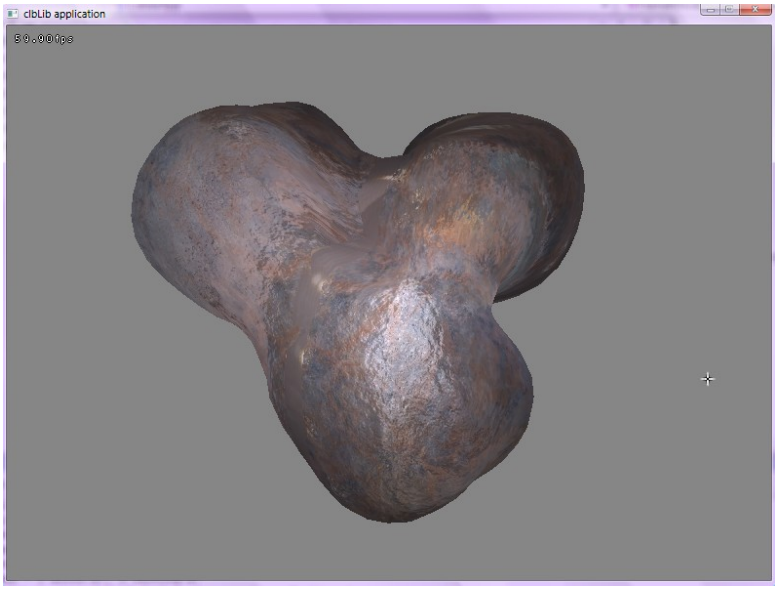




quantized to 4-bit palette.
The algorithm evaluates alpha correctly.

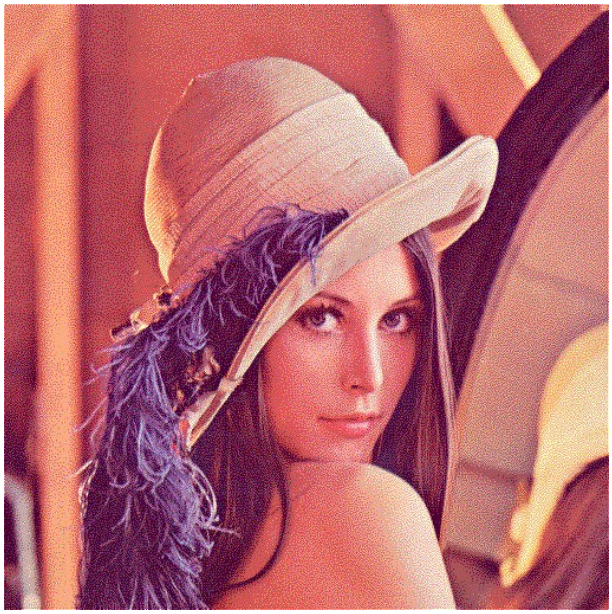
original

quantized to 8-bit palette with median cut



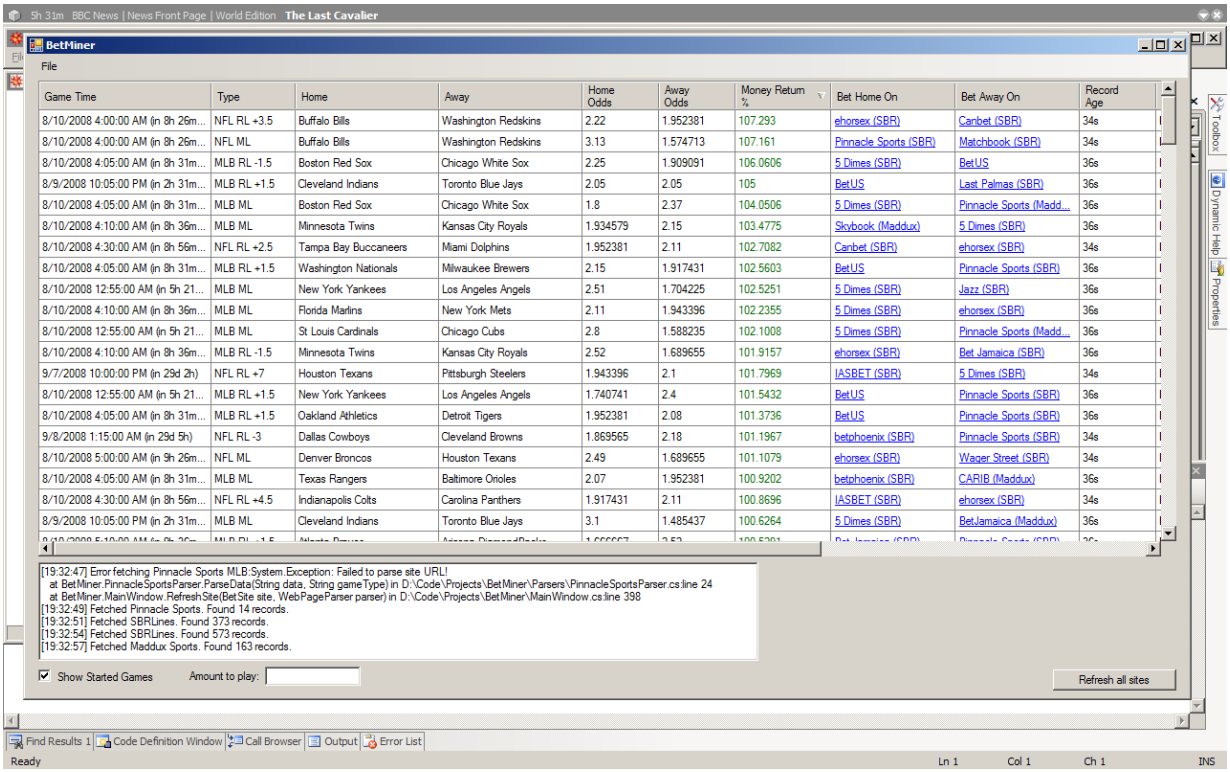
… coded real-time Marching Cubes (an isosurface to polygonal mesh -converter)...

True-color



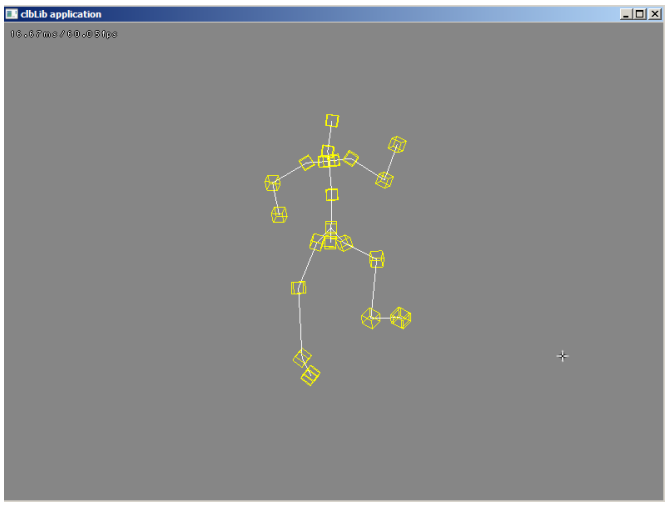
RGB222 (2 bits per color channel)

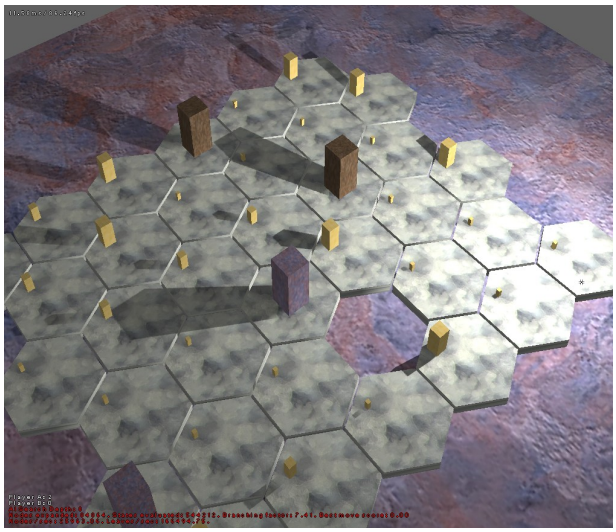
… written a perception-aware dithering color quantizer....



… an automated sports bet arbitrager ...

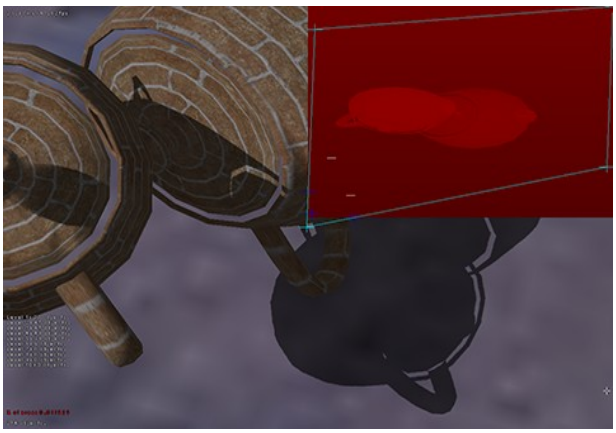
… a Cyclic Coordinate Descent, and a Jacobian Transpose -based IK solver...

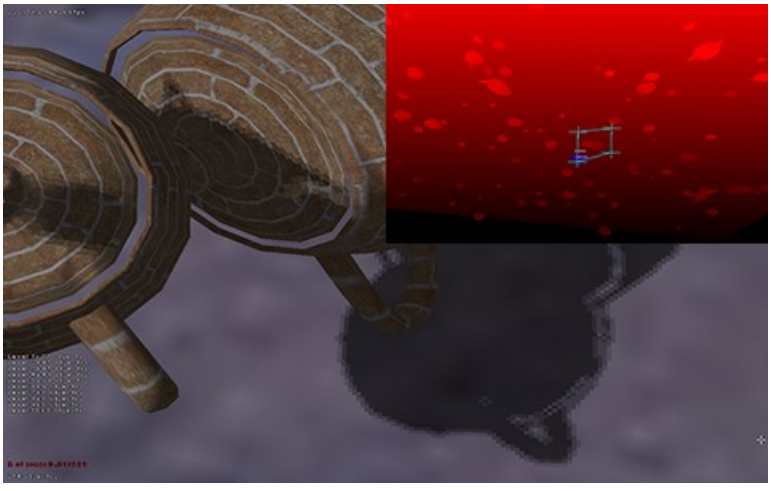


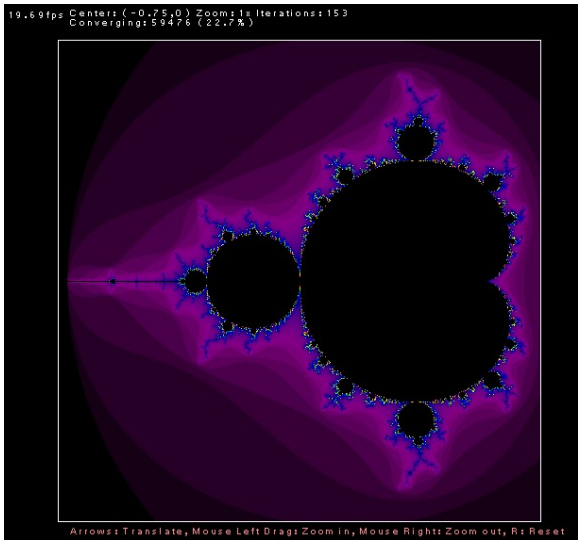


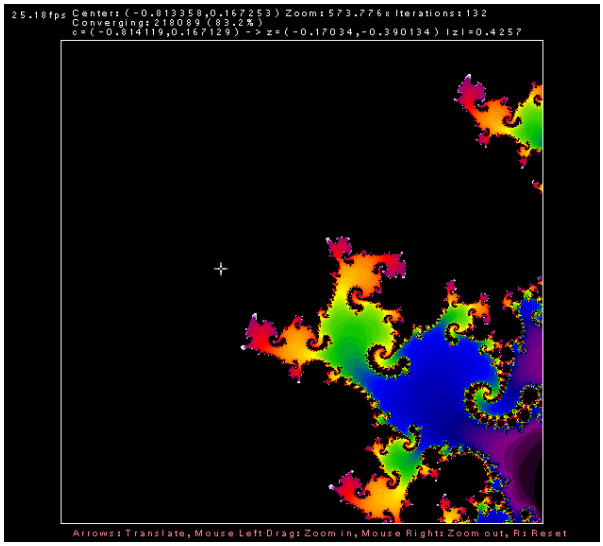
… a clone of the “Hey, That's My Fish!” board game with a Minimax AI ...

… and implemented methods for tight shadow caster frustum fitting...

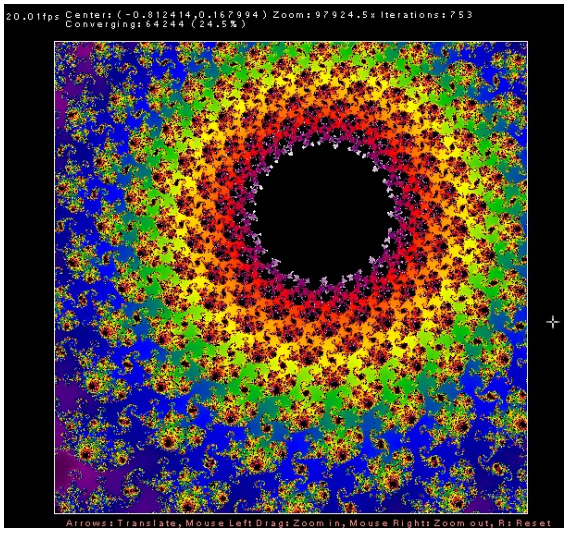


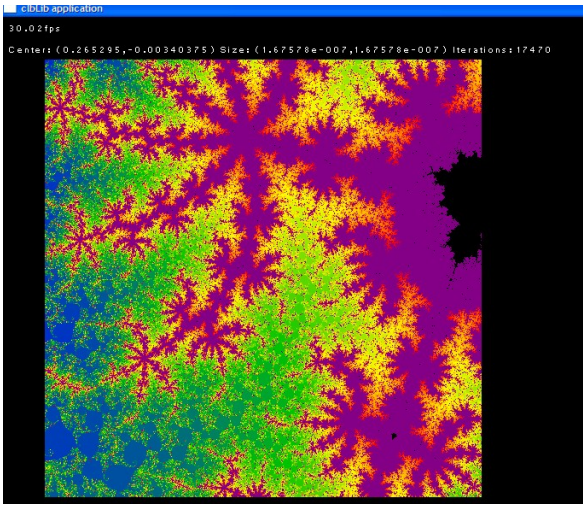


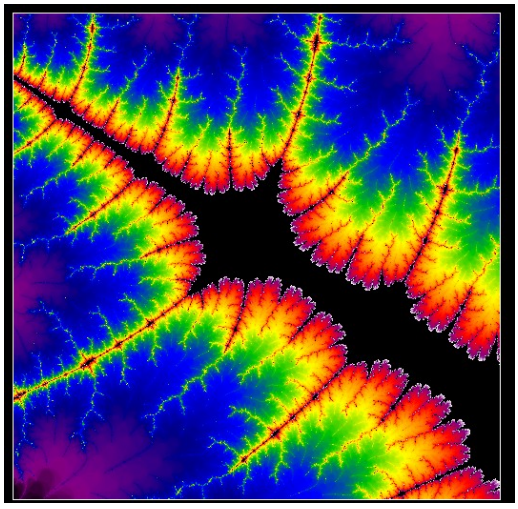


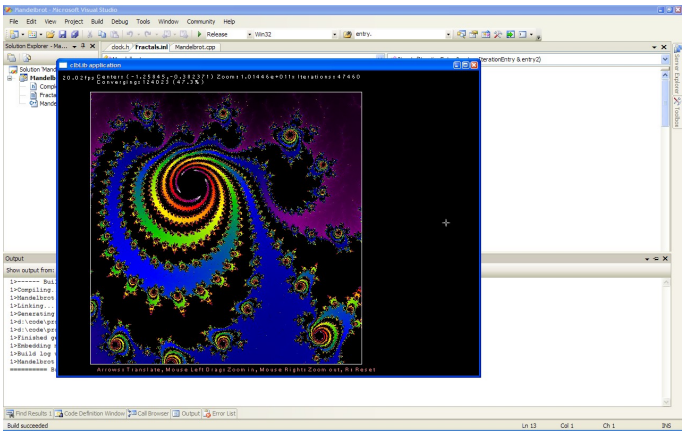


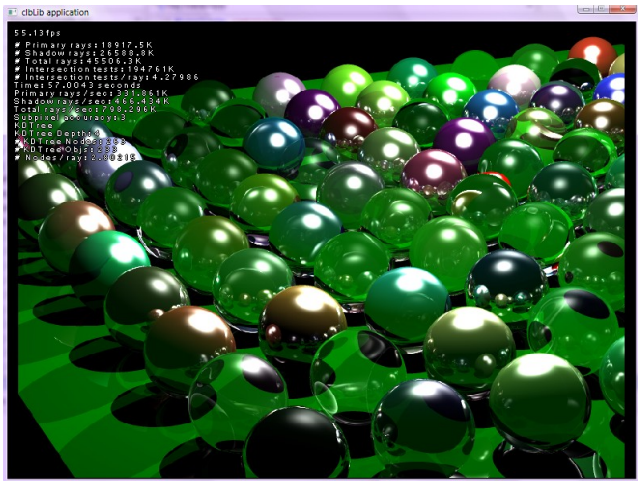
… developed a real-time SSE2-optimized Mandelbrot and Julia fractal viewer...

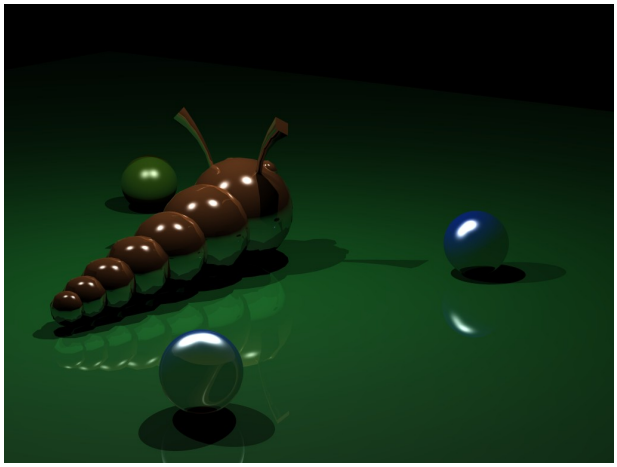




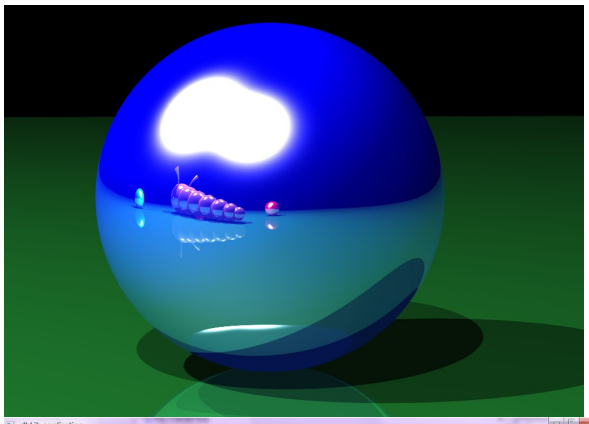


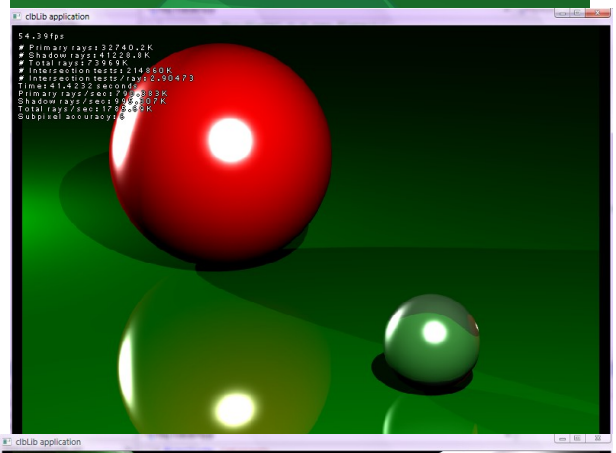


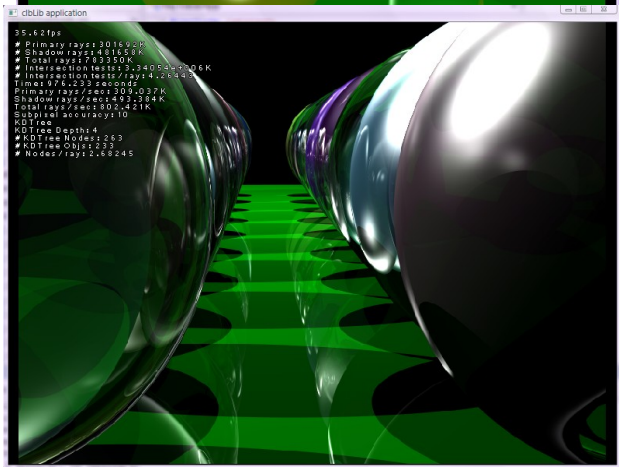




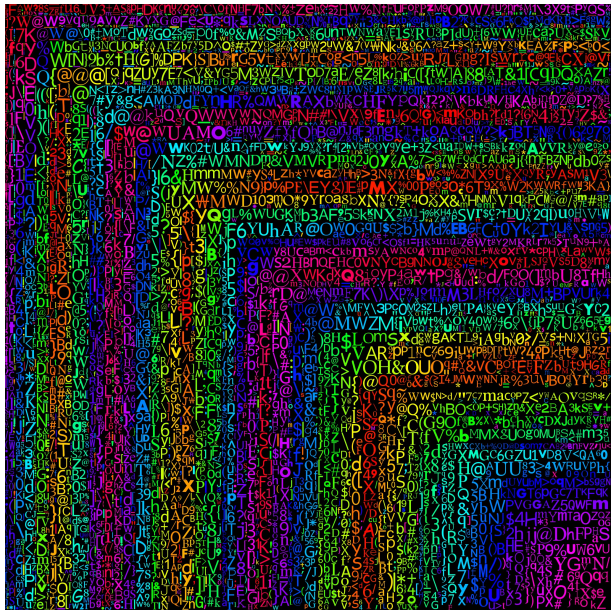
… and tried my hands at writing a real-time Raytracer, with kD-trees, and SSE optimizations. Performs at about 1M primary rays/sec on a dual-core laptop.







I wrote a paper on two-dimensional right-oriented rectangular bin packing. It was supposed to be my topic for my master's thesis, but sadly it was rejected. The algorithms in my paper are now used as a basis for several bin packers around the web:



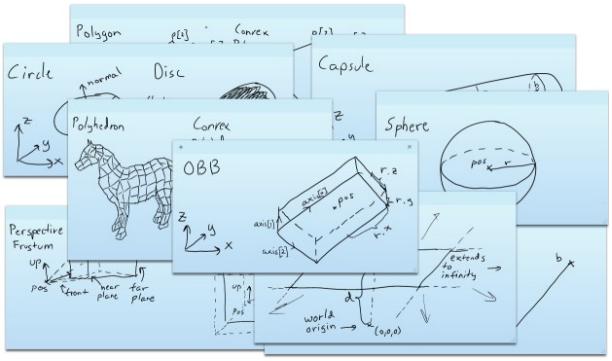
- freetype-gl
- SpriteMapper
- Bitmap Font Generator
- Urho3D Rendering Engine
- ps_scripts

and even in **two** commercial software packages: Zwoptex and Texture Packer Pro.

A series of blog posts and the original paper is available on my website: "Even More Rectangle Bin Packing".

I am the creator of the open source kNet networking library, which is a low-level network transport layer for streaming messages e.g. in games. It is written in C++, can be configured to run on top of TCP or UDP, and works on Windows, Linux and Mac.

- kNet home at Bitbucket.
- doxygen-generated kNet documentation pages.



Also, I am the author of MathGeoLib, an open source C++ library for matrix-vector math and primitive geometric object manipulation. The code is hosted at github repository juj/MathGeoLib.

Most recently, I have been working on an Asteroids-like game for the Tegra2 tablets, with Android NDK (C++), GLES2 and a strict 60fps-no-hiccups target.

And if it happens that I am not doing programming, I most often play Go at a local club. I hold an European Go Federation rank of 3 kyu, and I was the champion of the Lightning tournament at the London Open Go Congress 2010.

What would you rather see me do?

Let me know.