Jan Špaček

I am Jan Špaček. My experience includes:

- High-performance distributed systems
- Deep learning
- Web applications
- Desktop applications
- Physically-based rendering
- Programming language implementations

Work experience

Corona Renderer (Prague, 2017–2019)
Integration of Corona Renderer into ARCHICAD.
Bachelor thesis supervised by Jaroslav Křivánek, defended in 2018.
Written in C++ for Windows, using ARCHICAD APIs and Corona API.

Kiwi.com (Brno, 2016, 2017)
Design and development of a distributed intermediate storage for flights and development of distributed routing of flights (billions of flighs, a few terabytes of in-memory storage across many servers, millisecond latencies).
Design and development of a distributed system for high-performance and reliable loading of flights from the internal storage of Cassandra.
Written in C++ and Python with asyncio, using Cassandra, Redis and docker.

Bileto (Prague, 2015)
Development of an engine for routing in public transport networks fast enough to handle country-sized networks in real time using state-of-the-art algorithms.
Written in C++, using Redis and PostgreSQL.

Adash (Ostrava, 2014–2015)
Development of ADS, an application to visualize measured vibration data.
Design of efficient digital filters using advanced features of ARM processors.
Written in C++ with wxWidgets.

Selected projects

SkyGAN (2019–2020)
Generating high-resolution skydome images with deep learning (generative adversarial networks).
Master thesis supervised by David Futschik and Alexander Wilkie.

Dancerank.cz (2016–)
A database of results of dance sport competitions from several countries (including Czech, Slovak, Hungarian, Polish and international events).
Detailed analysis of competition results, ranking of couples, public API.
Predictions of future competition results with a novel machine learning model.
Advanced search with a custom text-search service.
Written in Python with asyncio, using MongoDB and Redis.

dort (2016–2017)
A physically based renderer heavily influenced by pbrt.

spiral (2015)
Full implementation of a dynamic compiled programming language: a compiler (in Rust), a runtime support library with garbage collection (in C++) and a standard library of basic algorithms and data structures (directly in Spiral).
Conducted as high-school final project; the standard library was later extended as part of a course at university.

…and a large amount of smaller projects, a few of them are on my GitHub @honzasp.

Skills

I am not limited to any particular language, platform or environment. I am most experienced in C++ and Python, but I also wrote Rust, Haskell, C#, Lua, JavaScript, Clojure, Ocaml, Go, Java, my own language Spiral, …

I happened to develop software mostly for Linux servers and Windows desktops. I (ab)used many databases (MongoDB, Redis, Cassandra, ...), countless libraries, frameworks, and APIs. I also enjoy low-level programming, such as bare metal Beaglebone and Arduino, CUDA, or SIMD programming with SSE/AVX. My editor is Vim.

I have solid background in computer science from Matfyz:

- Bachelor (Bc.) (2015–2018, with honors): general computer science
- Master (Mgr.) (2018–2020, *not yet finished*): artificial intelligence

Beside computer science, I dance Latin on a competition level (with my partner we have the highest national class “A”). I am also an avid reader in English and Czech.

Contacts

Web: `https://honzasp.github.io`
GitHub: @honzasp
LinkedIn: Jan Špaček