



Sergey Zhukov

FULL-STACK JAVA DEVELOPER · SENIOR DESKTOP DEVELOPER

Saint Petersburg, Russia

☎ (+7) 911-226-4992 | ✉ zhukovsd@gmail.com | 💻 zhukovsd | 📧 zhukovsd@gmail.com

Education

ITMO State University

B.S. IN INFORMATION SYSTEMS AND TECHNOLOGIES

Saint Petersburg, Russia

2007 - 2012

Skills

Programming languages	Java, Javascript, Delphi, C++
Web	HTML5, ECMAScript6, LESS, Bootstrap
Databases	MySQL, MSSQL, SQLite, MongoDB
Tools and Services	Git, SVN, Bash, PowerShell, Continuous Integration (Travis, GitLab CI)

Experience

iTVA

SENIOR DESKTOP DEVELOPER

Saint Petersburg, Russia

2010 - 2017

- Lead developer on a number of projects:
 - Native Windows applications
 - Cross-platform applications for Windows and MacOS
 - Web services
 - Libraries for internal usage within a team (internationalization library, persistent settings library)
- Was responsible for development and release of more than 10 software products
- Used technologies:
 - Programming languages: Delphi, C++, Javascript
 - UI frameworks: VCL, Firemonkey, Skinbuilder, Sciter, CEF
 - Libraries and databases: libtorrent, bass, MySQL, FireDAC, SQLite

Upwork

FREELANCER

Remote

2016 - present

- Profile - <https://www.upwork.com/freelancers/01fe466a51a96a2b6a>
- Average project rating - 5.00/5.00
- 20 jobs in different areas:
 - Java
 - Javascript/jQuery
 - Windows and MacOS installers
 - Various desktop projects: launcher and installer for MMO game, custom Chromium build, visual components for Delphi VCL

VBrand

FULL-STACK JAVA DEVELOPER

Remote

Mar. 2017 - present

- Implemented a screen capturing system (PowerShell, FFmpeg)
- Implemented a client-server application for video processing (cutting/merging/converting). RESTful API; Backend - Java, FFmpeg; Frontend - Javascript, Bootstrap, jQuery

Self-motivated projects

Maze generator

[GitHub project](#)

JAVA, GRAPH THEORY, SERVLET API

2015 - 2016

- Web application - <http://zhukovsd.github.io/maze-generator/>
- Algorithm works with planar graphs, which allows building mazes of arbitrary shape
- Finished generator was used during the implementation of maze solving Android game

Multiplayer minesweeper on infinite grid

[GitHub project](#)

JAVA, MONGODB, CONCURRENCY, WEB SOCKETS, HTML5

2016

- Web application - <http://5.101.123.222:8080/online-minesweeper/>
- Backend - JavaEE application, Frontend - HTML5/JS/Canvas, communication protocols - HTTP, WebSocket
- The grid is conventionally infinite and lazy-generated. A single unit is a chunk - an area of NxM cells
- A chunk is a unit of every internal interaction within the application:
 - Cells fetching from DB happens by indexed chunkID field
 - Fine-grained locking mechanism locks the field by chunks
 - Field area requesting by clients

Notable work projects

[Loviotvet.ru](#)

DELPHI

2010 - 2012

- Smart calculator for solving and simplifying math expressions and equations
- Self-implemented solving engine based on [reverse polish notation](#) builds a simplifications tree which allows displaying a solution in step-by-step fashion
- Supported math operations:
 - Common and decimal fraction operations
 - Polynomial operations including division
 - Solving linear and square equations
- A web service based on solving engine exposes RESTful API and serves requests from mobile clients and web application
- Freeware desktop, mobile and web clients were released for CIS market. Mobile clients have 500k+ installs, the service app served more than 12 millions of requests

An application for processing video content

JAVA, JAVASCRIPT, BOOTSTRAP

2017

- A client-server application for processing (cutting/merging/converting) video content. Typical usage is to create highlights clip from a show or to cut out ads from a broadcast record
- Architecture:
 - Backend - Java, FFmpeg
 - Frontend - Javascript, Bootstrap, jQuery
- Backend service exposes RESTful API which serves requests from web app UI views
- Supported video sources:
 - Direct URL
 - Torrent file
 - Youtube video URL

Plugin player

DELPHI, C++

2015

- Plugin which allows watching videos through p2p (torrent) protocol directly in browser
- The application consists of a "host" application which loads torrents and communicates with browsers. There are a number of standards for plugin implementation, so the application implements most popular ones:
 - PepperAPI (Chrome, WebKit-based browsers)
 - NPAPI (Firefox, pre-WebKit Opera)
 - ActiveX (Internet Explorer)
- "Host" application loads torrents using Delphi wrapper for libtorrent C++ library
- "Host" application communicates with plugins using Windows Messages
- UI for plugins is implemented with a Skinbuilder UI framework which is built on top of WinAPI