

---

## Experience

### Yandex

- Nov 2016 – **Lead Software Engineer**, Moscow.  
present Development of Yandex.Browser's rendering engine.
- Sep 2014 – **Senior Software Engineer**, Moscow.
- Oct 2016 Development of Yandex.Browser's rendering engine.
- Jul 2013 – **Software Engineer**, Moscow.
- Aug 2014 Development of Yandex.Browser's rendering engine.

I am a member of a team developing Yandex.Browser (<https://browser.yandex.com/>) – a Chromium-based web browser which is the second most popular browser in Russia on desktop and 4-th on mobile devices. Since its release in 2012 it has attained 19% share on desktop, 4.7% on mobile and has more than 11 million daily users.

Yandex.Browser is a huge project that contains millions of lines of code spread over thousands of files. It is cross-platform and is based on Chromium, which has a multi-process multi-threaded architecture. Chromium is written mostly in C++, but uses many other languages, including Objective C for Mac- and iOS-specific code, Java for Android-specific code, Python and Bash for tools, Javascript and HTML for internal pages.

Over the course of three years at Yandex I've participated in the development of almost every feature concerning Yandex.Browser rendering engine.

#### My key accomplishments as a software engineer:

- Fully rewritten media placeholders for Turbo mode, which required a huge refactoring spanning multiple subsystems of the browser, and supported the feature ever since.
- Implemented a substantial part of PushToCall service, which allows users to start a call from their mobile phone by clicking a phone number on a webpage in their desktop browser. I have a patent pending for the algorithms used in phone number marking.
- Implemented multiple prototypes concerning page layout and painting.

#### My key accomplishments as a senior software engineer:

- Participated in the development of Yandex.Alpha, which was a prototype of a revolutionary new browser, whose features were later incorporated into the main browser. I worked on browser chrome transparency and was responsible for the transparency API.
- Implemented partial table selection feature, which required a deep understanding of layout and painting mechanisms of the browser.
- Achieved a 3x performance improvement of in-page morphology-aware search.
- Was a lead programmer for the new native ad blocking feature.
- On my own initiative implemented a system for repository statistics visualization (similar to [https://github.com/tomgi/git\\_stats](https://github.com/tomgi/git_stats), but much faster and with some features that are meaningful only for Yandex.Browser repository). The system is written in Python using pandas library for data transformation. It generates lots of statistics and its visualization as

webpages, using Bootstrap framework, d3.js and Highcharts data visualization libraries. The system is used to detect suspicious differences between Yandex.Browser and Chromium and helps developers monitor their own productivity.

- Landed multiple patches to Chromium, including the implementation of partial link selection with Alt key.
- Implemented multiple JavaScript APIs.
- Conducted regular code reviews.
- Conducted numerous coding interviews.

#### **Technologies used:**

- *Languages:* C++ (STL), Python, JavaScript, Bash, Assembly.
- *Testing:* Google Testing framework and Google Mock library.
- *Source control:* git.

### [Lianozovo Electromechanical Plant](#)

Aug 2010 – **Software Engineer**, Moscow.

Jun 2013 Development of software for air traffic control systems.

Lianozovo Electromechanical Plant is a factory that builds air traffic control systems. I worked at the department responsible for software for AT controllers.

#### **Most notable projects:**

- A library for determining the location of an aircraft relative to AT sectors and predicting its position according to flight plans, weather and other conditions. This was essentially a library for determining an intersection of certain types of 3D objects. The library was written in C++ using Qt4 library.
- An application to visualize the work done by the aforementioned library. The application supports Windows and Linux and uses Qt4 library for GUI and OpenGL for 3D visualization.
- A replicated server application receiving information via ASTERIX (All Purpose Structured Eurocontrol Surveillance Information Exchange) protocol. The leader election algorithm was employed for the designation of the master server. The application is cross platform and was written in C++ using Qt4 library for networking.

#### **Technologies used:**

- *Languages:* C++ (STL, Qt4, OpenGL), SQL, Bash.
- *Testing:* Qt Test.
- *Documentation:* tex, dolphin.
- *Source control:* svn.

### [SimplyCeph](#)

Jul 2012 – **Remote C++ developer**.

Feb 2013 Automated modification of STL(STereoLithography) models.

During my work at SimplyCeph I implemented a tool that takes a model of an upper/lower human jaw made by a 3D scanner, removes unnecessary parts and searches for holes in the model. The unnecessary part removal is quite a challenging task that cannot be solved using precise methods, so I had to develop a heuristic algorithm using A\* graph path finding algorithm to achieve the required precision.

### Technologies used:

- *Programming languages*: C++ (STL, Boost.Graph), Bash.
- *Testing*: Qt Test.
- *Source control*: svn.

---

### Technical skills

- *Programming languages*: C++, C, Python
- *Libraries*: STL, QT
- *Operating systems*: Linux, Mac OS, Windows
- *Version control*: git, svn
- *Other*: gdb, lldb, vim, bash, tex

---

### Education

2005 – 2012 **Specialist degree**, *Moscow State University, Department*: mechanics and mathematics, *Specialization*: applied mathematics.

---

### Languages

Russian	<b>Native</b>
English	<b>Upper intermediate</b>
Czech	<b>Upper intermediate</b>