

Ivan Kotenkov

M +420777001876
E ivan.kotenkov@gmail.com

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 one of the most popular browsers in Russia with more than 11 million daily users.

While at Yandex, I have participated in the development of almost every feature concerning Yandex.Browser rendering engine, with some occasional work in other areas such as GUI and infrastructure.

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.
- Implemented multiple prototypes concerning page layout and painting.

My key accomplishments as a senior software engineer:

- I worked on browser chrome transparency and was responsible for the transparency API Yandex.Alpha, a prototype of a new browser whose features were later incorporated into the main browser.
- 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.
- Implemented a system for repository statistics visualization (similar to https://github.com/tomgi/git_stats, but much faster and with some features that are meaningful only for Yandex.Browser repository). 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.

My key accomplishments as a lead software engineer:

- I have been reviewing most of the Yandex.Browser modifications to Blink rendering engine and was responsible for its code health.

- Worked on improving the image decoding pipeline.
- Implemented animation throttling for energy saving mode.
- Implemented layout switching for in-page search.
- Provided consultations about Service Worker implementation to the developers of a game store embedded into the browser and implemented parts of the store dealing with offline.

Technologies used:

- *Languages:* C++ (STL), Python, JavaScript, Java, Bash, SQL, Assembly.
- *Testing:* gtest, gmock, junit.
- *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.

Education

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