

Ivan Ogloblin | Curriculum Vitae

Education

Saint-Petersburg State University

Sept 2019 - July 2023

Bachelor of Science in Computer Science and Software Engineering

Related Coursework:

- | | | | |
|-----------|--------------------|-------------------------|-----------------------|
| ○ C++ | ○ Algorithms | ○ Mathematical Analysis | ○ Quantum Computing |
| ○ Kotlin | ○ Machine learning | ○ Discrete Mathematics | ○ Quantum Information |
| ○ Python | ○ Unix | ○ Statistics | ○ JavaScript |
| ○ Haskell | ○ Operating system | ○ C# | ○ html and css |
| ○ Scala | ○ Algebra | ○ Data Bases | |

Programming experience

Yandex developer intern

July - Sept 2021

Worked in two commands on backend c++/python/sql. Developed support system for training scripts to work with an optimized structure for storing variable logs. Wrote tests for components, that were used to prepare data for recommendations neural network. Got acquainted with the concepts of services and levers. Plunged into the intricacies of communication between services and systems for transmitting information with errors for debugging.

Huawei assistant engineer, developer

October 2021 - January 2022

Worked on backend C#/.netASP/EntityFramework/Autofac + frontend 3js/react/VR. Developed system of package communication with no delay, that alternates between http and signalR requests.

Did research work on handwriting recognition using convolutional network under "Human Computer Interactions". Got familiar with CNN, RNN and LSTM structures.

Projects

Smashy Ninja

2018

- <https://play.google.com/store/apps/details?id=com.PixArt.Pouc>
- <https://github.com/StudioShader/Smashy-Ninja>

Back in highschool I made a mobile game with Unity 3d engine, published in Google Play, you can play it right now!

Archiver

2019

- <https://github.com/StudioShader/huffman-archiver>

Used Huffman algorithm in implementation for data compression and decompression.

You can run it now, with any C++ compiler.

DoNotExplode

2019

- <https://github.com/StudioShader/DoNotExplode>

Procedurally generate self-intersecting path for ball to bounce with a certain rules. An example of a billet for one of my game ideas. With an implementation of an interesting algorithm that I developed.

ML-projects

2019

- <https://github.com/StudioShader/ML-Projects>

I included implementation of Ant-colony and Genetic algorithms for "Travelling salesman problem". Also contains realization of K-means, SVM, Clustering and neural network algorithms. Just python scripts here

RTV-redactor

2020

- https://github.com/makselivanov/RTV_redactor

As a course project I wrote an algorithm that is able to recognize different handwritten geometric shapes (square circle rhombus) without using any machine learning techniques. I used ideas of interpolation angles and point structures.

Programming skills

- C++, Python, C#, C, Java, JavaScript, html, CSS, Kotlin, Haskell, Scala, SQL, Lean
- ASPnet, EntityFramework, Microsoft Sql express, React, three.js
- Git, Linux, Unity3D, SVN, Blender(3d modeling), protobuf, Shiny.

Quantum computing experience

Term paper

2021

I did a semester project on the topic “quantum algorithms for VRP and VRPTW (Vehicle Routing Problem with Time Windows) problems” with application to the real case problems of building the routes for drilling machines for oil production in collaboration with GazpromNeft – one of Russian major oil companies. I was directly assigned the task of studying current best practices for solving logistics problems on classical computers. The next step was to study current results on solving this problem by quantum and quantum-inspired methods. As many optimization problems, this one turned out to be possible to reduce to QUBO (quadratic unconstrained binary optimization) and solve it using quantum optimization algorithms such as VQE and QAOA. Then I was to develop (program) a simple solver of the multi-traveling salesman problem for small-scale problems (toy problem). It can run locally on a simulator from Qiskit. During this work, I perfectly understood the intricacies of launching and testing quantum algorithms using simulators and at IBM cloud system, how to look for quantum-inspired algorithms and test their applicability. (unfortunately I cannot share any code because of the privacy regulations of GazpromNeft)

Courses

2021-2022

- Course on introduction to quantum computations: Grover's algorithm, Deutsch–Jozsa algorithm, quantum permutations, quantum Fourier Transform, quantum search, Q-RAM, Shor's algorithm.
- Course on quantum information: density operator, noise in quantum systems, closeness of quantum states, quantum correction codes and their realization, classical and quantum entropy, bandwidth of quantum channels, transmission of quantum information over a noisy quantum channel, quantum cryptography.
- Additional seminar with the 'GazpromNeft' team: Phase estimation algorithm, QAOA algorithm, QAA algorithm, VQE algorithm, q.search as q.simulation, black box algorithm limits, speed up of NP-complete problems, q.search optimality, q.search in unstructured database.

Achievements

ICPC

2020

- 41 Place, Northwestern Russia Regional Contest St.Petersburg, October 26, 2019
- Honorable Mention, Northwestern Russia Regional Contest St.Petersburg, 14 November, 2020

Open olympiad

2018

- Top 60 out of 1100 in "Open olympiad in Mathematics" 2018 and 2016
- Top 174 out of 1404 in "Open olympiad in Mathematics" 2017
- Top 109 out of 1103 in "Open olympiad in Physics" 2018

International scientific school conference "XVIII Kolmogorov Readings"

2019

- I took [third place](#) in the discipline of computer science and mathematical modeling

Languages

Russian (Native), English (Upper-Intermediate)