Artsiom Kh.
Senior Senior C++ Developer

SUMMARY

- Software Engineer with over 7+ years of experience designing and delivering robust software solutions across diverse industries, including manufacturing automation, robotics, audio processing, UAV control, and communication systems. - Proficient in C++ (C++11, C++17, C++20) with a proven track record in developing both backend and frontend systems using React Native, Qt (QML, Qt Widgets), and PHP-CPP. - Skilled in creating high-performance multithreaded servers, implementing WebSocket protocols, and integrating cutting-edge technologies such as io_uring and advanced signal processing algorithms. - Demonstrated expertise in deploying solutions for real-time audio processing, cloud-based parking systems, and 3D modeling of industrial components. - Experienced in robotics and sensor integrations (Riftek, Nanotec, Rozum, Basler SDKs) for automating precision tasks in manufacturing. - Adept in database management (MySQL), DevOps (Docker, Bazel, Git), and cross-platform development on Linux, Windows, Android, and iOS.

TECHNICAL SKILLS

Main Technical Skills	C++ (7 yr.), CMake (5 yr.)
Programming Languages	C++ (7 yr.), JavaScript (3 yr.), PHP (1 yr.)
Android Libraries and Tools	Android APIs
JavaScript Frameworks	three.js
JavaScript Libraries and Tools	three.js
Data Analysis and Visualization Technologies	Talend
SDK / API and Integrations	Android APIs, Pusher, Windows API
Third Party Tools / IDEs / SDK / Services	CMake (5 yr.), Microsoft Visual Studio Code, Qt Creator (2 yr.), Visual Studio
Virtualization, Containers and Orchestration	Docker (2 yr.), Oracle VM VirtualBox
Operating Systems	Linux (2 yr.)
Scripting and Command Line Interfaces	Make (1 yr.)
UI/UX/Wireframing	STL

Mail / Network Protocols / Data transfer	WebSockets
Other Technical Skills	ACF, Borland, Qml (4 yr.), SDK, widget, WSL2

WORK EXPERIENCE

Software Engineer, ADAPTIVE NOISE SUPPRESSION SYSTEM

04.2024 – till now

Summary: Developed a real-time audio processing system that applies noise suppression to digital audio streams, using C++ for processing and React Native for the user-facing front end. The system dynamically adjusts noise levels in real-time to ensure crystal-clear audio quality. A key feature is the custom voice activity detector (VAD), which intelligently distinguishes between voice and background noise, optimizing suppression parameters for seamless operation.

Responsibilities included developing the audio processing core, integrating it with a React Native frontend, and ensuring system reliability through rigorous testing and bug fixes.

Responsibilities:

- Developed the real-time audio processing engine in C++, incorporating advanced signal processing techniques for noise suppression and voice activity detection;
- Built a React Native frontend for cross-platform user interaction, providing intuitive controls and real-time feedback;
- Designed and implemented a custom VAD for dynamic noise adjustment, significantly improving audio clarity and user satisfaction;
- Conducted performance optimization and debugging to ensure seamless integration between the C++ backend and the React Native frontend;
- Collaborated with audio engineers to ne-tune suppression algorithms and validate system performance under various environmental conditions.

Technologies: C++, JavaScript, Linux, Bazel, Git, React Native, Android, iOS, STL, Android Studio.

Software Engineer, GROUND CONTROL UNIT

Duration: 03.2023 – 04.2024

Summary: The Ground Control software is a versatile solution for managing and overseeing unmanned aerial vehicle (UAV) operations. Its user interface (UI) is highly customizable, allowing users to tailor it to their specific operational requirements. The software supports seamless integration with various UAV communication protocols, enabling compatibility with various UAV platforms and communication standards. Through its intuitive interface and flexible features, the software empowers users to control and monitor UAVs effectively, ensuring efficient and safe operations in diverse environments.

Responsibilities:

- Successfully implemented new enhancements in the project;
- Successfully migrated from Qt5 to Qt6;
- Made changes in deploying for docker for Qt6;
- Made work estimates;
- Participated in bug xing and code review processes;

- Regular addition of new features and prompt bugs;
- Provided expert consulting on software architecture and integration strategies;
- Created docs with information on how to configure the environment for development in Qt Creator;
- Collaborated with the Scrum Master and Product Owner to refine the product backlog for upcoming sprints.

Technologies: Linux, C++17, CMake, Qt, GitHub, BitBucket, Docker, QtWidgets, QML, VirtualBox, Qt Creator.

Software Engineer, CAR PARKS PAYING SYSTEM

09.2022 – 03.2023

Summary: I was working on the backend of a web portal for admins and facilities owners to make it possible for them to create/edit/manage tenants and tenant admins for them.

I also worked on the cloud for mobile apps, changing the calculation of the parking fee according to a new validation feature that I and the team created. I worked mostly on the C++ part of the web portal backend and cloud (PHP-CPP, MySQL) and less on the PHP and JS part.

My tasks were implementing new features and fixing bugs. Also due to the situation with the project I made building and deploying of C++ extensions for PHP for docker and CentOS 7 where the biggest problem was to make all things work for old projects and to nd and install all libs for CentOS 7 and docker.

Responsibilities:

- Successfully implemented managing tenants and tenant admins on the web portal backend;
- Successfully implemented validation logic (deducting from parking fee amount of validated hours) on the cloud;
- Fixed bugs and provide a code quality improvement for team members;
- Provided business consulting to clients, analyzing parking facility requirements and proposing tailored payment solutions;
- Completely made building and deploying of C++ extensions for PHP to production;
- Took a part in creating new tables and changing old tables in the existing MySQL database.

Technologies: CentOS 7, C++11, Make, JSON, MySQL, GitHub, JS, HTML, Docker, PHP-CPP, PHP 7.2, WSL2, Visual Studio Code.

Software Engineer, MESSAGE BROKER SERVER

03.2022 – 09.2022

Summary: I was responsible for developing a high-performance backend server written in pure C++ running on a Linux environment. This server was designed to implement the WebSocket protocol, utilizing cutting-edge io_uring technology for the asynchronous processing of client requests. This project presented an exciting opportunity to explore the latest innovations in C++ and Linux, ultimately contributing to developing a robust and efficient communication platform.

Responsibilities:

- Successfully integrated the WebSocket server into an existing message broker, enhancing the system's real-time communication capabilities;

- Created a multithreaded WebSocket server that significantly improved the server's responsiveness and concurrent client connection handling;
- Utilized io_uring technology to optimize client connections and data exchange, resulting in exceptional system performance and low-latency communication;
- Collaboratively estimated tasks, prioritized work items, and planned sprint cycles with the team, ensuring efficient project management and goal alignment;
- Conducted daily meetings with the team to maintain clear communication, resolve issues promptly, and achieve productive collaboration throughout the project.

Technologies: Linux, C++20, Make, BitBucket, WebSocket, Postman, Visual Studio, WSL2, epoll, io_uring, Pusher.

Software Engineer, BRAKE DISK GEOMETRIC PARAMETERS CONTROL SYSTEM

08.2021 – 03.2022

Summary: I was responsible for a multifaceted system controlling brake discs' geometric parameters. This system encompassed a range of components, including laser scanners, laser sensors, step motors, and FullHD cameras. It played a crucial role in ensuring the quality and precision of brake disk production, particularly in monitoring key parameters like width and curvature.

Responsibilities:

- Through meticulous data collection, I contributed to the system's ability to maintain high precision in monitoring brake disc parameters, thereby improving the overall quality control process;
- Implemented efficient data processing algorithms. I streamlined the system's ability to transform raw data into meaningful insights, reducing processing times and enhancing the accuracy of assessments;
- The automation of CSV report generation not only saved time but also increased the consistency and reliability of reporting, making it a valuable tool for quality assurance;
- Collaborated departments resulted in a system that could readily adapt to evolving requirements, ensuring its ongoing effectiveness in monitoring and controlling brake disc parameters.

Technologies: C++17, CMake, QT, OpenCV, GitLab, QML, QT Widgets, QCustomPlot, QT Creator, Nanotec SDK, Riftek SDK, Basler API, Win API.

Software Engineer, FACTORY DETAILS CHECK SYSTEM

03.2021 – 08.2021

Summary: I spearheaded the development of a comprehensive desktop application tailored to automate the quality control process in manufacturing. This project focused specifically on checking the geometric parameters of factory details. It was a testament to my end-to-end expertise, from gathering initial customer requirements to delivering the finalized application.

Responsibilities:

- Through meticulous data collection and efficient processing, the application became a valuable tool for automating the assessment of geometric parameters, thus contributing to precision and efficiency in factory detail quality control;
- Implemented customizable detail thresholds allowed users to adapt the application to various factory requirements, enhancing its versatility and usability;
- Regular addition of new features and prompt bug xing demonstrated an agile approach to software development, keeping the application current and robust;

- Maintained active communication with the customer, leading to a product that seamlessly met their requirements, fostering customer satisfaction and long-term collaboration.

Technologies: C++17, CMake, QT, GitLab, QML, Qt Creator, QThread, MVP, Riftek SDK, Win API.

Software Engineer, AUTOMATED SEAM WELDING SYSTEM

08.2019 – 03.2021

Summary: In my pivotal role, I contributed to developing a sophisticated system to automate seam welding using a robotic arm equipped with a laser scanner. This groundbreaking project leveraged robotics prowess and advanced technology to achieve real-time welding trajectory planning in two planes and generate a comprehensive 3D model of the welded seam.

Responsibilities:

- Participated in the development of a system that facilitates real-time welding trajectory planning and the construction of 3D models of welded seams, I contributed to the automation of seam welding processes, revolutionizing efficiency and precision in manufacturing,
- Successful collaboration with another company on the robotic arm's integration showcased my ability to work harmoniously across organizational boundaries, resulting in a seamless and integrated system;
- Translated the robotic arm's API from Python to C++ optimized performance and provided a more versatile and efficient tool for robotic arm control;
- Committed to diligent debugging and bug resolution played a significant role in the application's reliability and stability, ensuring smooth and trouble-free operation;
- I efficiently generated 3D models using a company's SDK, enhancing the system's capacity to provide in-depth analysis and visual representations of welded seams.

Technologies: C++17, CMake, QT, GitLab, QML, QT Widgets, QT Creator, QThread, Riftek SDK, Rozum SDK.

Software Engineer, LASER SCANNER CERTIFICATION APP

03.2018 – 08.2019

Summary: I led the development of a utility application designed to streamline the certification process for the company's laser scanners. A standout feature of this project is that it was a solo endeavor, with me taking on the full spectrum of tasks. The resulting application can gather critical data from laser scanners, perform intricate pattern analysis, and generate certification reports in PDF format. Additionally, it features an intuitive user interface that allows for process control, complemented by a live visualization widget, providing real-time insights into the scanner's perspective.

Responsibilities:

- Crafted a user interface that is intuitive and user-friendly and enhances the application's accessibility, making it approachable for users of varying technical backgrounds;
- The introduction of user guides offered an invaluable resource for users, promoting effective utilization of the application and contributing to a smoother onboarding process;
- Quick implementation of new features showcased my agile approach to development, making it easy to adapt the application to changing requirements;

- Diligent bug xing underscored the application's reliability and minimized potential disruptions.

Technologies: C++, CMake, GitLab, Borland C++, RAD Studio, Riftek SDK.

EDUCATION

Computer Science and Software Engineering