(Soft)Overlord100

Put photo of the mobile platform here

Or how to create a Robotics project in just 7 weeks, when half of your teammates are not Robotics

Overview

The idea for the project was suggested by Innopolis Industrial Robotics Laboratory. We goal was to create a **mobile platform** capable of following user commands, with friendly UI and a virtual simulation, showcasing the functionality.

Our team was responsible for the backend part of the project. That is, we had to create:

- SLAM, Path Planner, Controller
- Database
- APIs for Frontend and Hardware/Simulation

Market Research

Similar products exist both in Russia, China and Europe/USA, however, all of them have disadvantages for global Russian market.

Mobile platforms from Russia are developed for specific companies only, and are not available for global market.

Mobile platforms from China lack in support, and cannot be easily modified.

Mobile platforms from Europe/USA are simply not accessible in Russia.

Competitive Advantages

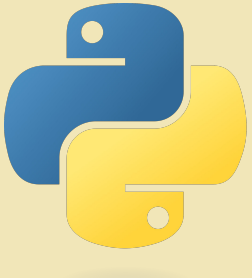
- Solution can be used in different environments
- Cheap cost, compared to competitors
- General purpose
- The functionality can be showcased via virtual simulation, not only shown on the video with actual robot
- Additional functionality can be added, such as cameras, robotic manipulators, etc.

Put photo of the simulation here

Implementation

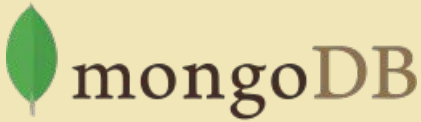
We decided on the following technological stack:

- C++ - used for developing ROS2 nodes
- Python - used for database and Frontend-Backend API
- ROS2 Humble
- Docker
- MongoDB



In the beginning, we planned to implement the following:

- SLAM
- Path Planner
- Controller
- Database
- Behavior Tree
- Sensors Fusion



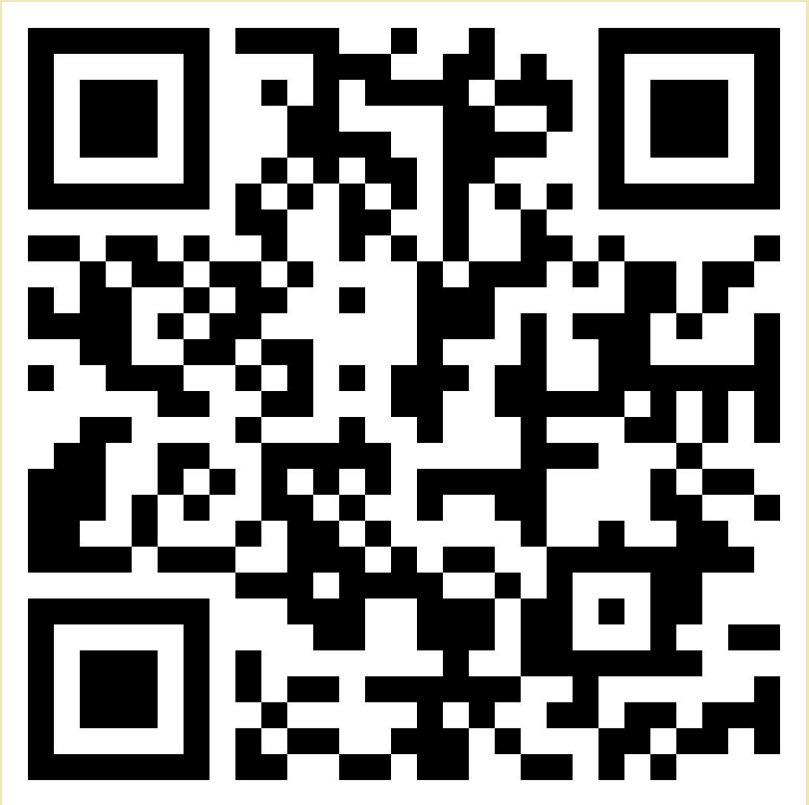
Architecture of the project

Scheme with project structure

SLAM

To create SLAM node, we used an open-source solution, slam_toolbox. Its integration in our project was smooth, however, we had to merge lidar data, since two lidars are used in our robot.

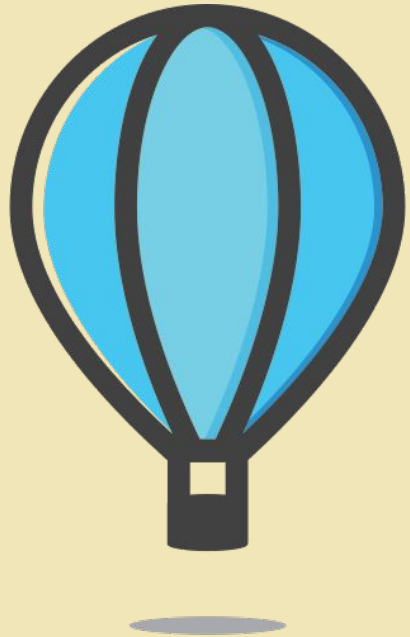
Link to GitHub for slam_toolbox



Path Planner

For Path Planner an open-source solution, Nav2, was used, since it is designed to work together with slam_toolbox. Unfortunately, we faced issues with this node, since we could not fix problems with odometry frame for a long time.

Link to NAV2 documentation



NAV2

Controller

We wrote our own implementation for Controller node, since existing packages are not suitable for our mobile platform, as certain information is computed before transferring to backend. We faced issues with connecting controller to simulation, since simulation team did not understand in what measurement units the velocity was given.

Data management and API

For Data management, MongoDB, a NoSQL database was used. We faced some issues with integrating database as a ROS Node.

Frontend-Backend API was created using rosbridge-server.

- How API works (rosbridge_server)
- Note about collaboration with other teams
- What information is exchanged
- Problems faced

Link to demo

If possible, otherwise video of demo

Testing

We have tested both separate parts of the project (comparing the results the gave us with expected output), and the project as a whole. You may see the videos of test launches here.

Github Workflow

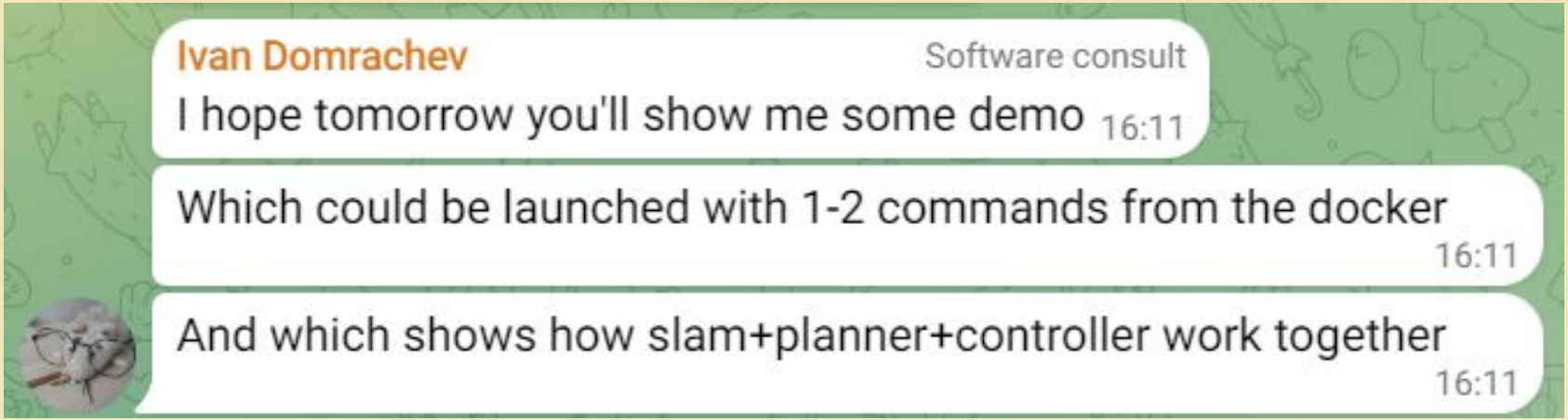
- Note about Pipeline

Documentation

We have created a documentation of our project in Notion. It contains information about different parts of the project, used tools and instructions how to launch the project.

Link

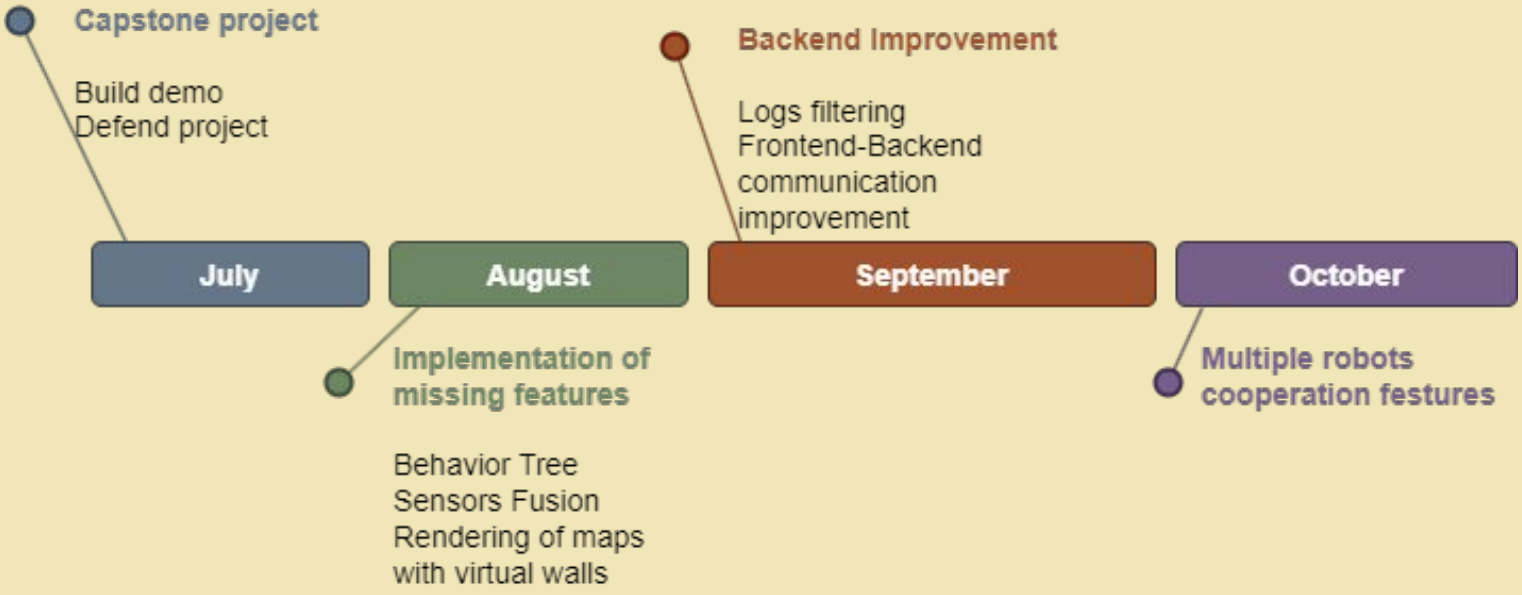
Feedback



We established weekly meetings with our mentor, Ivan Domrachev, to get feedback on our progress. Moreover, we gathered feedback from fellow students.

Future plan

- Improve backend
- Implement robots cooperation



Conclusion

We have created the backend containing the following components:

- SLAM
- Path Planner
- Controller
- Database
- APIs

While working on the project, we have gained new skills, learned to overcome problems and communicate when problems arise. But we will not stop here. We still have ways to improve our project, and we all hope to continue the productive work.

Our team

Ekaterina Mozhegova (lead)
Iurii Podkorytov
Mukhammadrizo Maribjonov
Yehia Sobeh
Ali Hamdan
Anastasiia Shvets
Saveliy Khlebnov

Add photos and roles

Acknowledgments

- Robotics lab
- Ivan Domrachev
- Karim ElDakroury
- Other people we took examples for testing from

Add photos and comments

Contact info

Link to Github
Link to Innopolis Robotics Society
Maybe some additional contact information, like email or telegram