1 Introduction

This is the first report from the NacionalnaKlasa team, outlining the progress we have made on the project since the competition kick-off. Since the competition began, our team has made major progress in a short period of time. It's important to mention that we received the car about a week ago, but in the meantime, we familiarised ourselves with the code, worked on algorithms, and ran simulations. After that, we focused more on the car itself. For this report, we have completed the initial project setup, analysed the provided documentation and codebase, established the simulation environment, successfully started and remotely controlled the car, and initiated work on lane detection, traffic sign detection, and hardware inspection.

2 Planned activities

These are our currently agreed arrangements and are subject to change in the future, based on how each of us performs in our respective fields of responsibility.

- Konstantin Malešević

Konstantin is our team leader and is responsible for the regular and continuous work on our project. He is quite familiar with neural networks, so he is focused on forming a particular model and training the model on a specific dataset of traffic sign images, which has been previously analysed in detail.

- Minja Drakul

Minja's work is closely related to Konstantin's work and they work together on traffic sign detection. Minja is in charge of finding datasets, preparing the video for the first report and filtering out the noise after Iva has finished her part of the work. She is prone to branding our BFMC project in the future.

- Filip Goldberger

Filip is in charge of organising the repository on GitHub for versioning and progress tracking. He is currently working on refactoring car servers and preparing simulations for testing behaviour in a controlled environment. He also has the most experience in hardware, so his future work will likely focus around embedded programming and fixing hardware.

- Iva Mančev

Iva is very familiar with image processing, so her work is focused on lane and edge detection. She found multiple edge detection algorithms, so she is now focused on evaluating and finding the best solution for their project. Due to her affinity for hardware and embedded, she will regularly collaborate with Filip.

3 Status of planned activities

- Reading the documentation, assigning the tasks and setting up GitHub — DONE
- Installing a physical environment for our car, including traffic lanes and traffic signs — DONE
- Starting the car and achieving remote control of the car — DONE
- Running and adjusting the simulation — DONE
- Researching and finding the best suitable algorithm for line detection — *in progress*

- Researching machine learning tools and making different datasets for traffic-sign detection *in progress*
- Resolution of a partially solved hardware problem *in progress*
- Comprehensive analysis of the given GitHub code *in progress*

4 General status of the project

- We decided to focus on simulation during this project because we didn't have a physical car. We spent a few days setting up the environment for simulation, and we also encountered some issues in the XML files, then the simulation was adapted for ROS2. After solving those issues, we are now able to control the car remotely in the simulation. In the next period, we are planning to use the simulation for testing algorithms and collecting training data.
- We successfully started the car, disassembled and reassembled it, and then restarted it again to test the remote control functionality. Based on the documentation, we inspected and cleaned the hardware.
- We identified several edge detection algorithms, two of which proved particularly promising. One performs better on straight lines but struggles with roundabouts, while the other handles curved lines more effectively but tends to approximate and generate a larger number of short line segments in the video. Both algorithms were tested on videos available in the documentation. In the coming period, we will test the algorithms in real conditions on the car and with data from the simulation to evaluate which of these approaches performs better.
- We are setting up a dataset environment using Roboflow, creating multiple dataset versions with preprocessing, augmentation, and train/validation/test splits from publicly available data. In parallel, we are evaluating resource-efficient object detection models, initially selecting YOLOv8n and currently testing YOLOv3n, YOLOv5n, and YOLOv12n. All models will be trained and tested on the same dataset, which will be refined iteratively if performance is unsatisfactory. Model comparisons will be conducted using videos uploaded to Mega, as the target track is not yet finalised (unless we record our own test data over the weekend).
- We are focused on solving a partially solved hardware problem where the link connecting the servo motor and the wheel mechanism is hanging from a rod connected to the differential.
- As part of the project, a Python logger was implemented to record events and monitor system operation, with logs displayed in the terminal and simultaneously saved to a log.txt file.

5 Upcoming activities

In the next period, we plan to work on:

- Refactoring the car's server modules to enable WiFi connectivity.
- Analysis of the BFMC 'Brain' code to understand its operation and implement our functionalities; and diving more into the given code on GitHub, as the project goes by.
- Implementing our basic lane detection on the car.
- Creating our algorithm for traffic sign detection and implementing it on a car.
- Begin testing the car in simulated conditions as well as in the physical environment.
- Further studying of relevant topics such as OpenCV and machine learning in order to ultimately make the car fully autonomous.
- We'll probably need to do more hardware testing to make sure everything is working properly.