

Reflection on the Zumi Robot Challenge

Mykhailo Andrusiak

Working on the Zumi robot challenge was a unique and rewarding experience. It helped me not only improve my technical skills but also understand the importance of communication, structured thinking, and resilience when things don't go as expected.

At first, the task seemed simple — make the robot move along a defined path. However, I quickly realized how complex this challenge was. From getting Zumi to drive straight along a line to tuning the size and distance of QR codes for reliable object recognition, I had to dive deep into the robot's internal logic and sensor data processing. One of the parts I'm most proud of was building a logging system that tracked every action in real time. This became our debugging tool, allowing us to understand what had already been executed and take decision for next steps.

Zumi felt like a game to me — I could spend hours testing and exploring her potential, often losing track of time. I really enjoyed experimenting with her functions and figuring out how she “sees” the world. Some solutions I found made me genuinely happy and proud.

Still, there were many challenges. The most frustrating part was stability. Sometimes the robot worked well, but the next day, without any obvious reason, the same code would behave differently. This made debugging difficult and unpredictable. We also faced hardware limitations — the battery didn't last long, and only one person could fully test Zumi at a time. We tried to work in parallel, but whenever two or more people attempted to connect to Zumi simultaneously, the connection became unstable. The only reliable solution was switching between Git branches and letting one person run tests at a time. To manage this, we divided the tasks into smaller parts and tested them individually, which helped us save time and avoid conflicts.

Teamwork was another big aspect. One of our teammates had difficulties keeping up with team expectations, and despite our efforts — including two structured coaching sessions and multiple attempts to clarify roles and goals — collaboration remained challenging. In the end, we made the difficult but necessary decision to split the team to maintain progress. This experience taught me how important open communication and shared accountability are — and that sometimes, even with strong effort, not every collaboration can be made to work.

What I also learned is how valuable it is to let team members take responsibility for their parts, and only give input when asked. I also learned to break down tasks into the smallest testable parts to reduce uncertainty during movement testing.

In the end, this project taught me how hardware and software interact, and how to stay adaptable when things go wrong. Even though it wasn't always easy, it was a meaningful experience that strengthened both my confidence and my curiosity.