

# Journal

## Contents

Journal .....	1
Monday (week 5) 29-01-2024 .....	1
Tuesday (week 5) 30-01-2024 .....	1
Wednesday (week 5) 31-01-2024 .....	1
Thursday (week 5) 01-02-2024 .....	2
Friday (week 5) 02-02-2024 .....	2
Bibliography .....	2

### Monday (week 5) 29-01-2024

---

#### Kristoffer

- At Beumer
- Started project
- Had meeting with Jonas discussing what to start with the first 2 weeks
- Read these papers:
  - A visual introduction to Gaussian Belief Propagation [1]
  - Distributing Collaborative Multi-Robot Planning With Gaussian Belief Propagation [2]
- Tried compiling examples from <https://github.com/aalpatya/gbpplanner> but faced issues with missing X11 headers, even though they were installed on my system.

### Tuesday (week 5) 30-01-2024

---

#### Kristoffer

- Worked from home.
- Created GitHub repository <https://github.com/AU-Master-Thesis/gbp-rs> as we want to rewrite the <https://github.com/aalpatya/gbpplanner> in Rust.
- Looked at different Rust simulation/visualization tools to use.
  - <https://macroquad.rs/>
  - <https://nannou.cc/>
  - <https://bevyengine.org/>
- Decided to go with **bevy** as it has a lot of community support/solutions and we thought its ECS system is really cool!.
- We read through the introduction book for bevy, to learn the core concepts behind the ECS paradigm and how applications are structured in bevy.

### Wednesday (week 5) 31-01-2024

---

#### Kristoffer

- At OrbitLab
- Continued to have issues compiling the code for <https://github.com/aalpatya/gbpplanner>.

- We both decided to re-flash our OS with NixOS.
- Spent some getting acquainted with the terminology and methodology of how to do things in NixOS
- Create a `flake.nix` for both our Rust port and `gbpplanner` to create a reproducible environment, where we can compile and run the code without issue.

## Thursday (week 5) 01-02-2024

---

### Kristoffer

- At OrbitLab
- Continued learning about NixOS and setting up our development environment, with the tools we like to use.
- Spent some time trying to port the code from [https://colab.research.google.com/drive/1-nrE95X4UC9FBLR0-cTnsIP\\_XhA\\_PZKW?usp=sharing#scrollTo=NzotHENoaY6g](https://colab.research.google.com/drive/1-nrE95X4UC9FBLR0-cTnsIP_XhA_PZKW?usp=sharing#scrollTo=NzotHENoaY6g) to our Rust implementation.

## Friday (week 5) 02-02-2024

---

### Kristoffer

- Worked from home.
- Continued our attempt to port the code from [https://colab.research.google.com/drive/1-nrE95X4UC9FBLR0-cTnsIP\\_XhA\\_PZKW?usp=sharing#scrollTo=NzotHENoaY6g](https://colab.research.google.com/drive/1-nrE95X4UC9FBLR0-cTnsIP_XhA_PZKW?usp=sharing#scrollTo=NzotHENoaY6g) to our Rust codebase.
- Jens wrote the code, while we both discussed how to port the Python code to Rust.

## Bibliography

- [1] J. Ortiz, T. Evans, and A. J. Davison, “A visual introduction to Gaussian Belief Propagation”, *arXiv preprint arXiv:2107.02308*, 2021.
- [2] A. Patwardhan, R. Murai, and A. J. Davison, “Distributing Collaborative Multi-Robot Planning With Gaussian Belief Propagation”, *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 552–559, 2023, doi: [10.1109/LRA.2022.3227858](https://doi.org/10.1109/LRA.2022.3227858).