**Milrem Robotics**

**2025 assignment**

Martin Maasik

13.04.2025

**Assignement description:**

**Objective:**

Develop an interface prototype for an Unmanned Ground Vehicle (UGV) system, employing Vue 3 alongside HTML, CSS, and TypeScript.

**Key Deliverables:**

1. **Vue 3 Project Setup:**

* Initialize a Vue 3 application incorporating HTML, CSS, and TypeScript

1. **Map Integration:**

* Incorporate a digital map from a provider and/or library of your choice (e.g., Google Maps or Leaflet).
* Display the UGV's location on the map, marking a predefined starting point as its initial location.

1. **Planning, Engine Control and Driving:**

* Implement a mock-up control mechanism to start the UGV's engine. This should be achieved through a visually distinct button at top
* right corner, featuring both an icon and text for clarity. The button must have grey background with 80% opacity, black borders with 2px
* width and white text color.
* Ensure the UGV can be maneuvered on the map using the keyboard's arrow keys for directional movement. User should not be able to
* move UGV if engine is not started. The controls required for prototype are:

Move UGV forward

Move UGV backward

Move UGV right

Move UGV left

* Add a simple Drive-To-Destination feature.

Long press on the map lets you add a waypoint

After adding a waypoint, popup appears with buttons:

Drive - Changes UGV current location to the waypoints location

Save - Saves the waypoint to a store/runtime variable

Discard - Discards the waypoint that was created

* Add a simple view which displays all the saved way-points. Upon clicking the saved waypoint in the waypoints list, you can issue

following commands in a popup:

Delete - Deletes waypoint from store/runtime variable

Rename - Lets you alter the name of waypoint

Drive - Changes UGV current location to the waypoints location

1. **Engine Status Notification:**

* When the UGV's engine is off and arrow keys are pressed, display a popup notification advising the user to start the engine.

**Suggestions:**

1. First implement everything that is required by the deliverable, then focus on refactoring, formatting and styling. Keep the UI simple but intuitive.
2. Make the implementation as modular as possible without introducing too much boilerplate, ensuring that developers working on this in the future

could re-use your components.

1. Note down what was the most time consuming/difficult part of the assignment for you
2. Proper documentation on how to set up and use the application
3. Comment any use of AI systems

**Submission:**

* The code-base to be uploaded to Github
* Code base to include with instructions to set-up and use the applicationREADME.md
* Code base to include a demo video file in root showcasing the application

**Solution:**

This project is a prototype for an Unmanned Ground Vehicle (UGV), built usint Vue, HTML, CSS and TypeScript.

**Technologies used:**

* Vue 3 - main frontend framework
* TypeScript - Strongly typed JavaScript
* HTML & CSS - UI structure and styling
* Leaflet - map rendering and interaction
* Vite - development server and build tool

**Getting started:**

1. Clone the repository:

git clone <https://github.com/mmeest/Milrem-Beserker.git>

cd Milrem-Beserker

1. Install dependencies:

npm install

1. Run the application:

npm run dev

**Features and functionality:**

Uses Leaflet to display digital map

UGV’s initial position is marked on the map with red marker(engine is turned off by default)

Engine control:

* A button at the top-right corner starts or stops the engine.
* When engine is off button text ‘START ENGINE’ with green icon. UGV marker is red (no movement)
* When engine is turned on button text ‘STOP ENGINE’ with red icon. UGV marker is displayed as UGV icon. (can be moved 8 directions)
* Button design: grey background (80% opacity), 2px black border, white text.
* UGV movement is disabled if the engine is off.
* Pressing arrow keys when engine is off triggers a popup warning.

UGV movement (8 directions)

Use numeric keyboard keys to move UGV:

* 7 - Move upper left
* 8 - Move up
* 9 - Move upper right
* 4 - Move left
* 6 - Move right
* 1 - Move down left
* 2 - Move down
* 3 - Move down right

Waypoint system:

Mouse right click creates new waypoint on the map(blue marker)

To ‘Saved vaypoints’ appears new waypoint with name of new waypoint and 3 buttons:

* Drive - drives UGV to the waypoint if engine is turned on
* Rename - renames the waypoint
* Delete - removes the waypoint

**Notes**

* The waypoint system uses runtime memory (no backend or local storage).
* UGV movement is visual and not linked to physical measurements or obstacles.
* UI is intentionally kept simple for prototyping purposes.

## **AI Use Disclaimer**

Some parts of this project, including component structure and documentation writing, were supported by AI assistance (ChatGPT). All logic and implementation decisions were reviewed and customized by the developer.