Midterm Report

Group Members: Nion Schürmeyer, Slaven Jevtić, Luka Levac

Autonomous Exploration

We use the explore_lite package for greedy exploration of the given environment and create 2D map using gmapping. When the robot stops moving for a minute, the map is saved and the exloration is declared finished.

For launching the exploration either launch `explore_building.launch` or `explore_museum.launch` in the package my `my_robot_control/launch`.

In the building environment the robot does not have a problem to create a complete map. In the museum environment it frequently gets stuck. We are not quite sure why but maybe the openess of the world (no walls surrounding everything) poses a problem for the explore algorithm.

Todo: Live Long Slam So far we only map once and are not able to adapt to changes in the environment.

Navigation

For navigation we are working in museum_guide package. We launch the bringup.launch and turtlebot3_navigation files which open Gazebo and RViz. The change in those files was that we set the inital position coordinates back to (0, 0). The goal going forward is to put the starting position back to (-7.0, -2.5) and automate the 2D pose estimation in RViz (do it programatically), which we've tried by publishing the new initial coordinates to /initialpose topic, but we couldn't debug why it did not work. After launching, we prepared a script that publishes the desired goal to move_base_simple/goal which moves the robot to the location. We prepared the code so it iterates through 3 locations, one after another because we are planning to expand that to move to location of each QR code and after to the locations defined by the QR codes. The problem here was, that the script was sending the goals too soon. We implemented a condition that waits for the status of the robot to not be moving towards the goal, but the /status topic needs too long to change from one status to another, so the script skips the second defined point. We need to find a different approach to check when the robot stops moving and it successfully arrives to the desired location.

Computer Vision

QR Code Reading

For QR Code reading we are using the `zbar_ros` ros package and `reading_image` node.

The **package** uses the Zbar qrcode reader library. It subscribes to the `/image` topic and constantly looks for a qrcode in the image frame. When a qr code is recognized the it's immediately read and the contents of the code are sent to the `/barcode` topic.

Node that we wrote is working with those two topics. The script of the node subscribes to the to the /barcode topic and works with that info. On new data entry from the qr reader, the node takes that text and deciphers what type of a command it is (DR, FWY or FW). After that the text is broken into a simpler format of only numbers and sent on a new topic `/qrComm`. The navigation node, then listens for commands from this topic.

We had some problems with the rgb camera on the turtlebot in the gazebo. There was a lot of crashing so in the end, we just used the `usb_camera` package for testing purposes as it was actually easier to just read the qr codes from the laptop camera. Other than that there was mainly just text editing and converting (text to number and vice versa) done with the text of the QR Code. Time for disinfection is also calculated from the power and energy.

An example of the command sent to the navigation node:

```
data: "2 330 3 2 4  1 5  6 3 "
```

- 2 -> second type of command (FWY)
- 330 -> Wait time in seconds
- 3 -> Three coordinates
- 2 -> First x coordinarte
- 4 -> First y coordinate
- 1 -> Second x coordinate
- etc.

In the next steps we plan to implement *people detection* with the opencv library so the robot will turn off the lamp when a person walks in to the frame.

For lauching the qr code reading:

```
    $ roslaunch qr_code_reading image_proc.launch
    $ rosrun qr_code_reading reading_image

    $ rostopic echo /qrComm
```

On each new qr code read (so different text than the previous one), one message will be sent on the */qrComm* topic. The text used in the QR codes is exactly the same as in the email we got. With new lines and everything. For now the code works for first two typed of messages just for testing purposes. If the same QR Code is presented, nothing will be sent to the topic.

In the video we uploaded, first all of the necessary files are ran in the terminal. After that in the video stream and in the console you can see that whenever a **new** QR code is shown into the camera, the command is sent to /qrComm. If the qr code is the same, nothing happens.