# FIRST ROBOTICS PROJECT

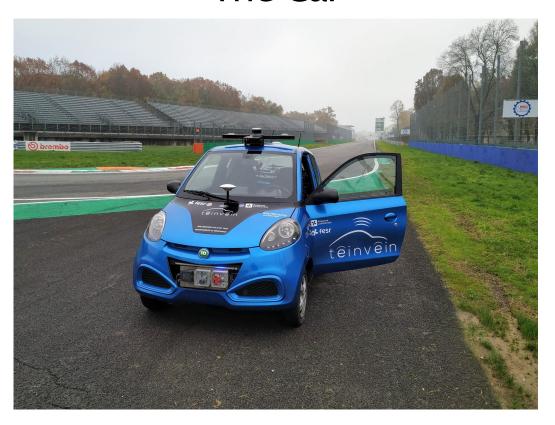
**ROBOTICS** 



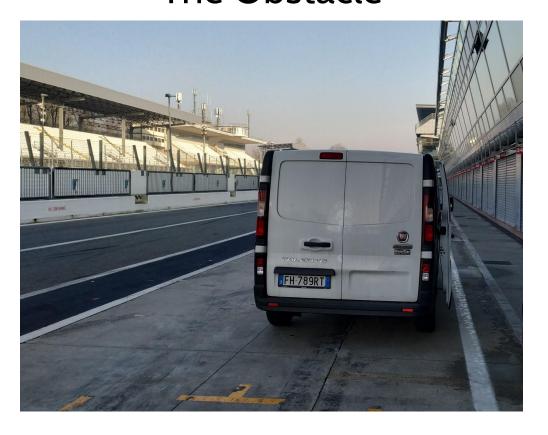
# THE PROBLEM



# The Car



# The Obstacle



#### **DATA**



Format: ROS Bag file

Data:

- Car GPS position: latitude, longitude, altitude
- Obstacle GPS position: latitude, longitude, altitude

# THE PROJECT



- Convert from lla to ENU, zero point should be a parameter in launch file (formulas for conversion are provided in example node)
- Publish TF and Odom for both car and obstacle (in ENU coordinates)
- Write a service that returns the distance between the two
- Publish a custom message with fields:
  - distance between car and obstacle
  - status flag
- Optional: use dynamic reconfigure to change status thresholds

#### LLA TO ENU



- You can use the ROS node provided as a starting point
- You need to run two instances of the node, one for each GPS source
- You need to specify ENU origin point with parameters in launch file
- Default value for ENU origin point is Car starting point
- Write all in a launch file (multiple node start and parameter set point)

# Compute distance

- To compute the distance use message filters to retrieve the last messages from both GPS sources
- Use the policy that best fit this scenario
- If GPS loose fix it returns (0,0,0). Handle this scenario when converting to ENU and then when providing distance.



### Custom message

- Node ask to the service the distance to fill the distance field
- Based on parameters (default value set in launch file) the flag value of type string assumes different values:

if dist > 5m: Safe

if 1m<dist<5m: Unsafe

if dist<1m: Crash

Parameters specify the three thresholds





Use Dynamic reconfigure to change the three thresholds of the custom message string field

Use three slider to change the values

#### **IMPORTANT NOTICE**



GPS data might disappear due to bridges, maybe only one will disappear.

Handle this scenario properly -> publish a distance of nan if the GPS data are not available, do not publish dist=0.0

All nodes should start with a launch file

Project can be done with an arbitrary number of node

#### TIPS FOR DEBUG



You can use rviz to visualize the odometry and tf topic

The bag represents a run in the Monza ENI Circuit

If you setup rviz properly you should be able to see the two vehicles moving on a trajectory which resembles the circuit layout

If the odometry (in meter) has too high values to visualize in rviz use a debug topic with odom=odom/100.0



# https://goo.gl/GonArW Project folder





- -Send **only** a tar.gz file (put the .txt file with info inside the archive)
- -Send via e-mail both to Simone Mentasti and Matteo Matteucci
- -name the e-mail "FIRST ROBOTICS PROJECT 2020"
- -Inside the archive:
  - -txt file (details next slide)
  - -folders of the nodes you created (with inside CmakeLists.txt, package.xml, etc...)
  - -do not send the entire environment (with build and devel folders)





#### File txt must contain (at least):

- -ID, name, surname of all team members
- -small description of the files inside the archive
- -name of the parameters
- -structure of the tf tree
- -structure of the custom message
- -description of how to start/use the nodes
- -info you think are important/interesting



#### Some more requests

Insert in the archive all the file you think are important, i should be able to properly recreate your workflow

Name the archive with your ID

Don't use absolute path

The project need to be written using c/c++ (no python code unless previously discussed)





Deadline: 8 May (3 weeks)

Max 3 student for team

#### Questions:

- -write to me via mail (simone.mentasti@polimi.it)
- do not write only to Prof. Matteucci
- -ask on Teams/Slack