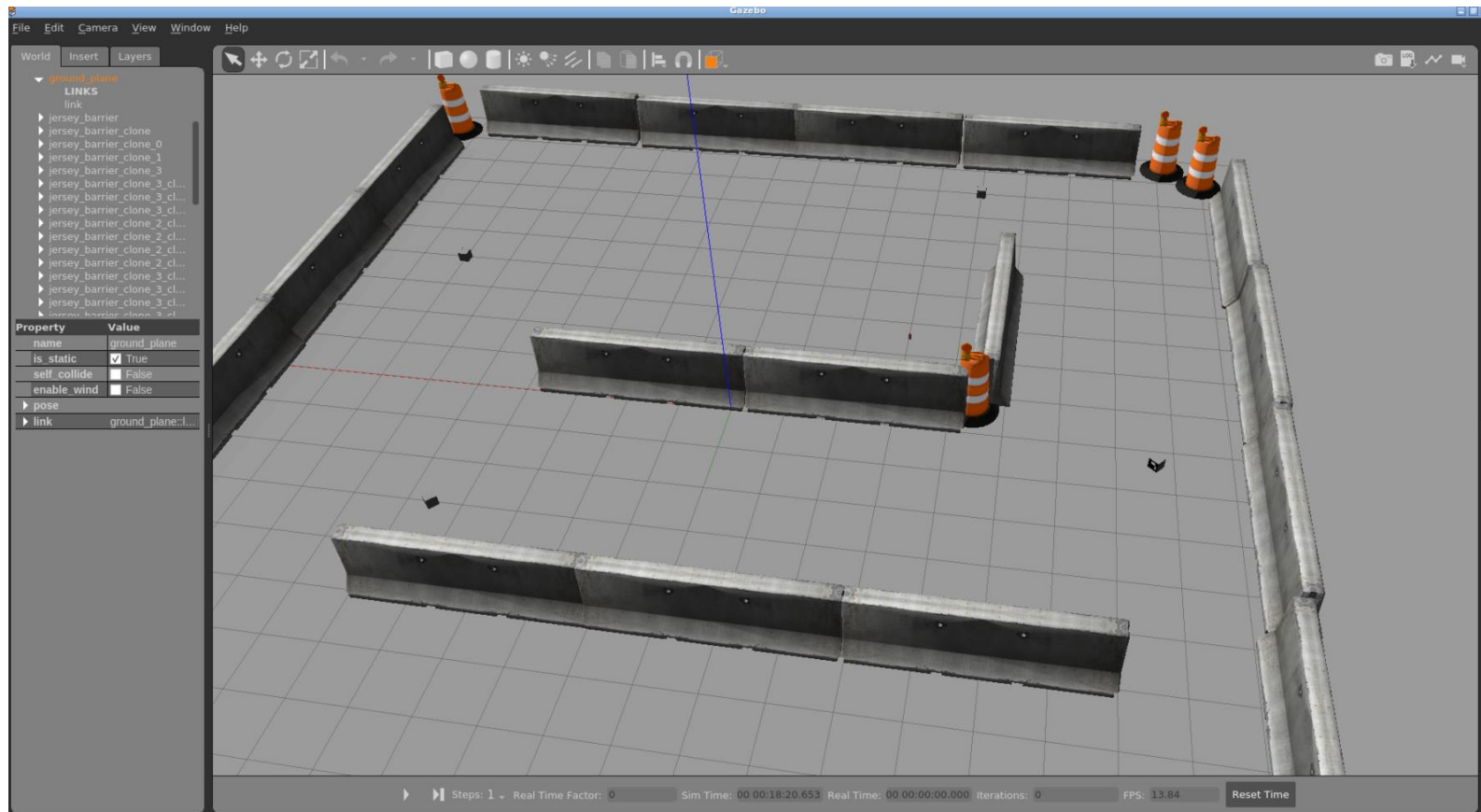


# 2<sup>nd</sup> Assignment

Carmine Tommaso Recchiuto

# Assignment 2

You have the following environment, with four markers in the environment, with IDs 11, 12, 13, and 15.



# Assignment 2

Develop a ROS package that lets a mobile robot endowed with a camera:

- find all markers in the environment
- go back to the initial position

However, you have some hints about the position of marker. You know that:

- marker 11 is visible from the position  $x = 6.0$ ,  $y = 2.0$
- marker 12 is visible from the position  $x = 7.0$ ,  $y = -5.0$
- marker 13 is visible from the position  $x = -3.0$ ,  $y = -8.0$
- marker 15 is visible from the position  $x = -7.0$ ,  $y = -1.5$

Implement the assignment both in simulation (the world file assignment2.world is given) and with the real robot.

You can implement the task in groups of 4-5 persons. Feel free to split the work, by having different persons working on certain aspects of the task (e.g. robot model, simulation, marker detection, ...)

Requirement:

- you should use ROSPLAN (or PlanSys2) to plan the actions of the robot

# Assignment 2

Working with the real robot, you can connect to your Rosbot by following this procedure:

- connect to the network TP\_LINK (Password is 03694008)
- login in ssh to husarion@<husarion\_ip> (192.168.1.xxx).
- [you can turn on the graphical interface as indicated in: https://husarion.com/tutorials/howtostart/rosbot---quick-start/](https://husarion.com/tutorials/howtostart/rosbot---quick-start/)

- once connected, run the tutorial\_2 launch file:

```
roslaunch tutorial_pkg all.launch
```

This will start the drivers of the camera and of the robot controller

- to run the aruco\_ros package directly on the rosbot, please use: [https://github.com/pal-robotics/aruco\\_ros/tree/melodic-devel](https://github.com/pal-robotics/aruco_ros/tree/melodic-devel) . Alternatively, you can run it on your pc by sharing the ROS master. Also, you can use the scp protocol to copy files on the rosbot.

# Assignment 2

- In simulation, you can spawn the rosbots in the gazebo environment ([https://github.com/husarion/robo\\_ros](https://github.com/husarion/robo_ros))

In ROS,

```
roslaunch gazebo_ros empty_world.launch  
roslaunch robo_bringup robo_gazebo.launch
```

- In ROS2 the Robo package should work only with the “humble” version and a specific. If you don’t have that version, but you want to use ROS2, feel free to use one of the other robots used during the class

# ROS2

The simulation part can be totally done in ROS2 (in case, using a different robot instead of the Rosbot)

Concerning the interaction with the real robot, since only ROS is currently installed on the robot, you can use the ROS-ROS2 bridge:

[https://github.com/ros2/ros1\\_bridge](https://github.com/ros2/ros1_bridge)

Please carefully follow the ReadMe if you are going to use it.

# Assignment 2

## Additional Requirements:

- Create a flowchart of your code, or describe it in pseudocode ([Pseudocode Examples \(unf.edu\)](https://www.unf.edu/~cs331/pseudocode-examples/))
- Add a video to your ReadMe, showing the behaviour of your code both with the real robot and in simulation
- Publish the new package on your own repository.
- **Deadline: 31 December 2023**

# Evaluation

- Code performance
- Code structure and clarity
- Respect of the requirements
- Organization of the repository (e.g., README in which you describe what the code does (possibly with flowchart or pseudocode), how to run the code, possible improvements, ... )