

SSY236

Assignment 02: Gazebo & Perception

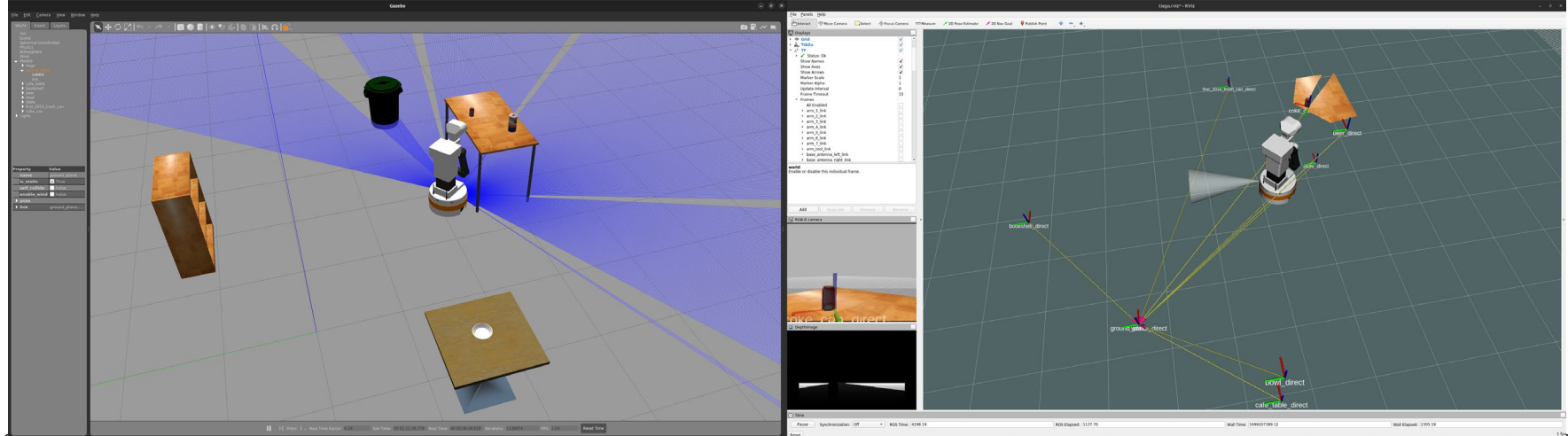
Karinne Ramirez-Amaro

Emmanuel Dean

Assignment 02

In this assignment, you can accumulate a maximum of 10 points

Goal: The robot should navigate its new environment and save the pose of the objects it sees.



Assignment 02 – PSL computers



- 1) Start Ubuntu
- 2) Create your workspace, similarly as we did in the last tutorials, e.g.
`ros/workspaces/ssy236_karinne`
- 3) Open the vscode

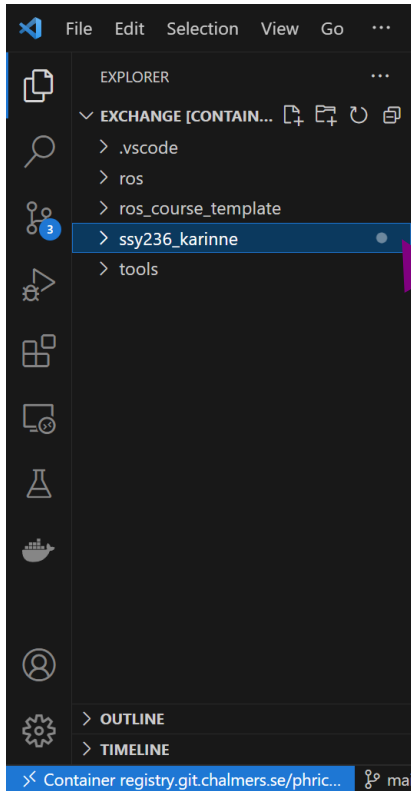
Assignment 02 – Docker system



- 1) Open Docker
- 2) Run the container
- 3) Open and configure Xlaunch
- 4) Open vscode
 - Attach the vscode to the container

Skip these instructions
if you are **not using**
docker

Assignment 02



- 1) Download the folder “assignment_2.zip” from Canvas → Assignment
- 2) Unzip the file which will generate a new folder. Inside this folder, you will find another folder called “world_percept_assig”
- 3) In your vscode, move to your workspace folder in the “EXPLORER” menu. Then, copy the folder “world_percept_assig” inside your src folder, for example:
 - ssy236_karinne
 - src
 - world_percept_assig

Assignment 02



- 1) Compile the new package “world_percept_assig”
- 2) Before compiling, make sure that you **source the right workspace**:

1) For the PSL computers do this:

```
cd ros/workspaces/ssy236_karinne/  
source ../tiago_public_ws/devel/setup.bash
```

2) For the docker system do:

```
source /tiago_public_ws/devel/setup.bash
```

- 3) Then, simply compile:

```
catkin_make
```

A screenshot of a ROS2 development environment. The top part shows a file editor with a URDF file named 'world_percept.urdf'. The file content includes a world named 'default' with a sun light source and various material properties. The bottom part shows a terminal window with the following commands and output:

```
root@8fb8b2fd427c:/home/user/exchange/ssy236_karinne# source /tiago_public_ws/devel/setup.bash  
root@8fb8b2fd427c:/home/user/exchange/ssy236_karinne# catkin_make
```

Assignment 02 – Task 1



Task 1 (2 points): First the robot should see all the objects displayed in Gazebo.

For this, you need to :

1) Modify the node called “direct_percept_node” to:

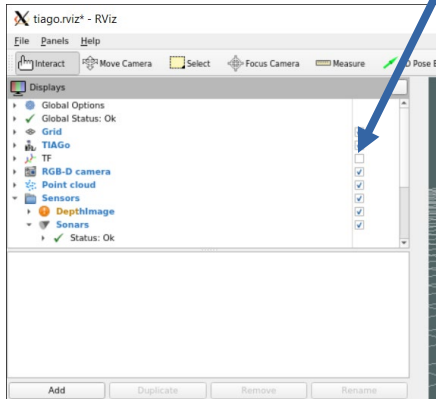
- Subscribe to the information from the gazebo using the topic (0.5 pts)
 - “/gazebo/model_states”
- Publish a TF broadcast of the information received from the subscriber

Look for the #TODO inside the file “direct_percept_node.cpp”

Hint: The information needed to run the node is already included in the CMakeLists.txt and package.xml

Assignment 02 – Task 1

Why is the TF important?



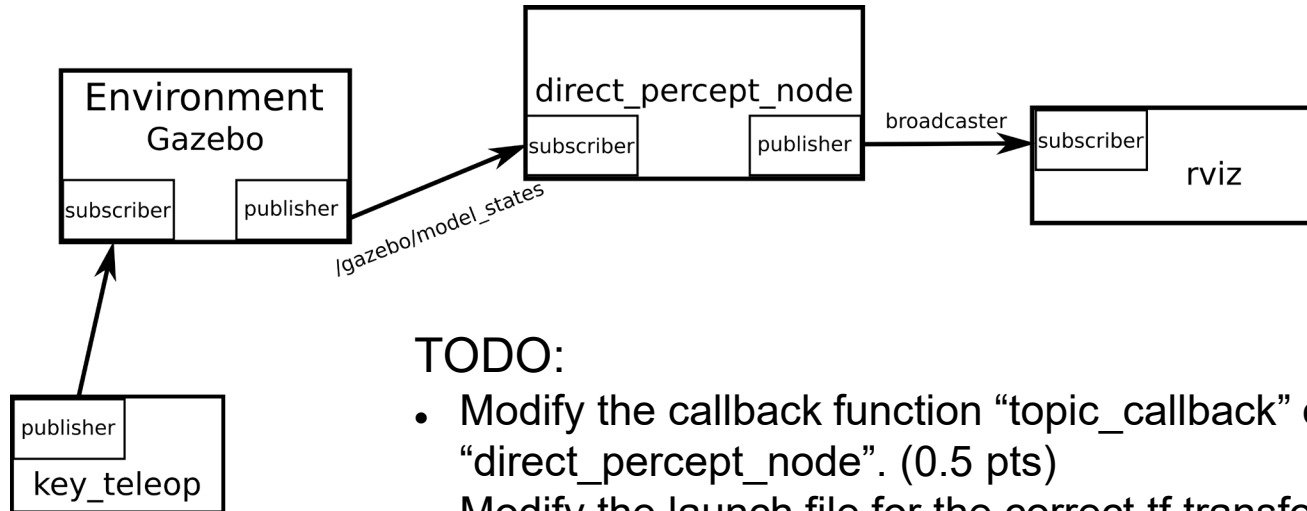
The robot has a **TF broadcaster** component that continuously broadcasts the transformation from

- the "base_footprint" to "base_link" to "base_sonar_01_link"
- and from "base_footprint" to "odom"

These transformations specify the relative position and orientation of the SONAR sensor with respect to the robot's base_link and the robot's base_footprint with respect to the global environment "odom".

Assignment 02 – Task 1

Expected outcome of Task 1 (2 points)



TODO:

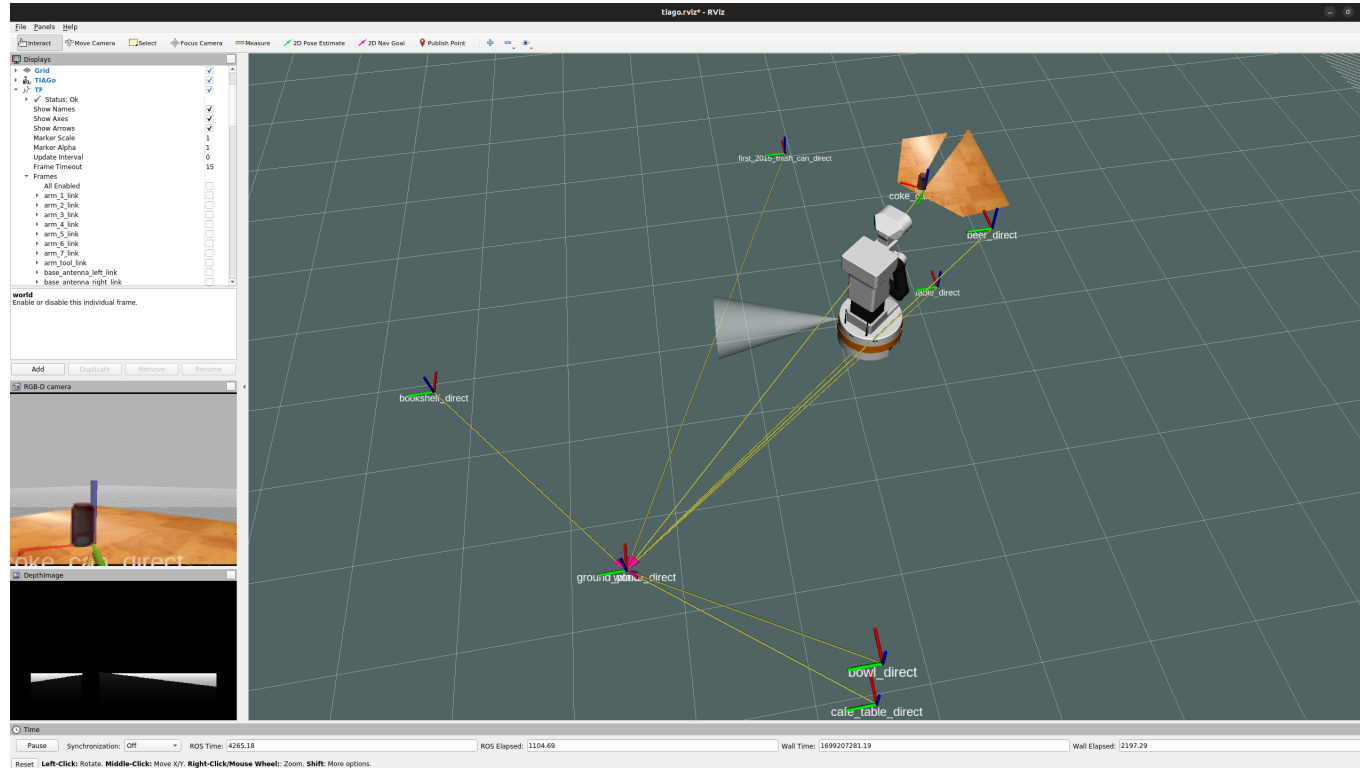
- Modify the callback function “topic_callback” of the “direct_percept_node”. (0.5 pts)
- Modify the launch file for the correct tf transformation (1 pt)

<https://wiki.ros.org/tf2/Tutorials/Writing%20a%20tf2%20broadcaster%20%28C%2B%2B%29>

Assignment 02 – Task 1

Expected outcome of Task 1 (2 points)

Hint: you can navigate to verify that the tf's are correct!

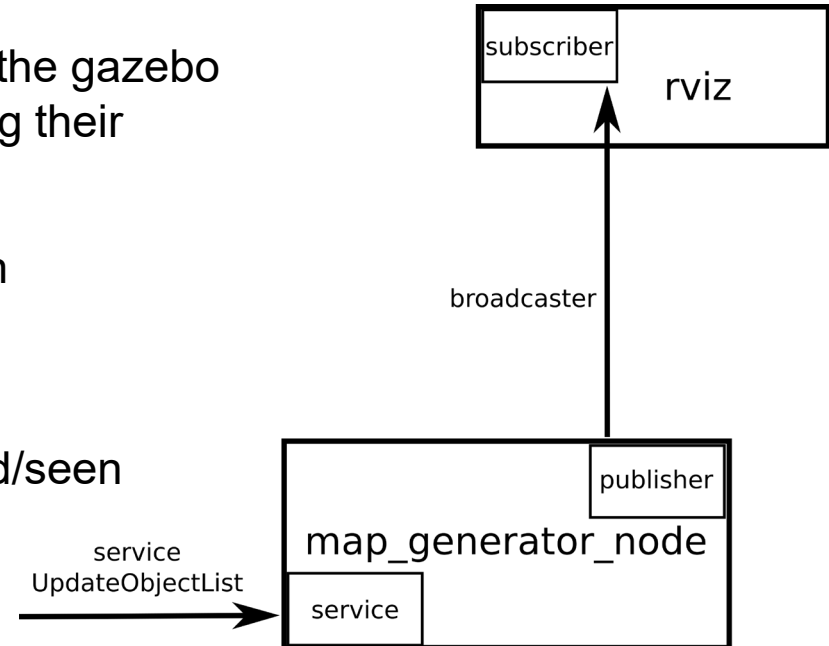


Assignment 02 – Task 2



Task 2 (3.5 points): The robot does not have access to the node “direct_percept_node.cpp” yet.

- This means that the robot needs to navigate in the gazebo environment and start finding objects and saving their information. ← Receive information
- We want the robot to save the information when requested, this means that we need to define a service/client node. ← Process the information
- Then, we want to display the TF of the identified/seen object. ← visualize information

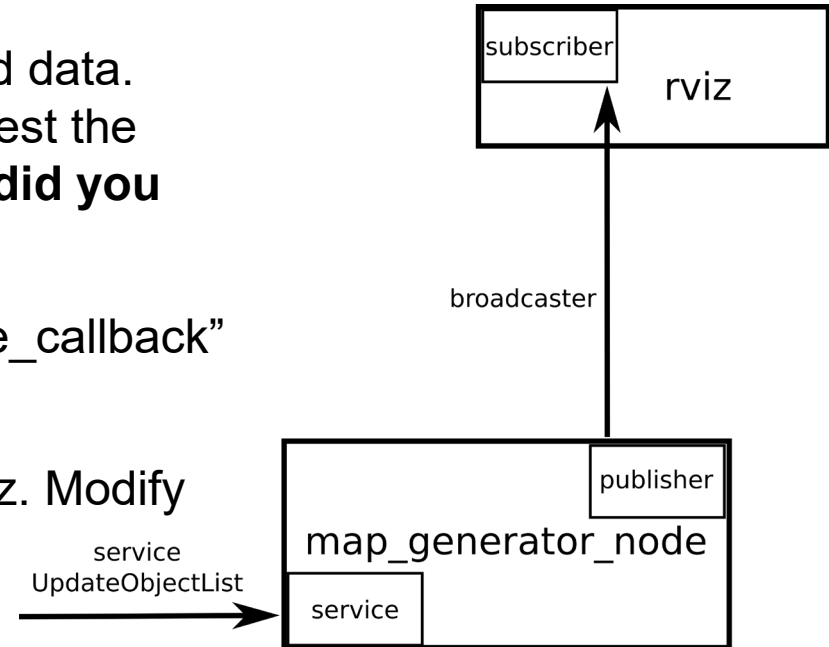


Assignment 02 – Task 2



First work on the file “map_generator_node.cpp” and fix the TODOs:

- First work on the service to store the received data. This means that you will use the terminal to test the “client” input. **Explain in a readme file how did you do it.** (0.5 pts)
- Fix the main parts of the function “srv_update_callback” (1 pt)
- Update the information that will be sent to rviz. Modify the function “tf_timer_callback” (2 pts)



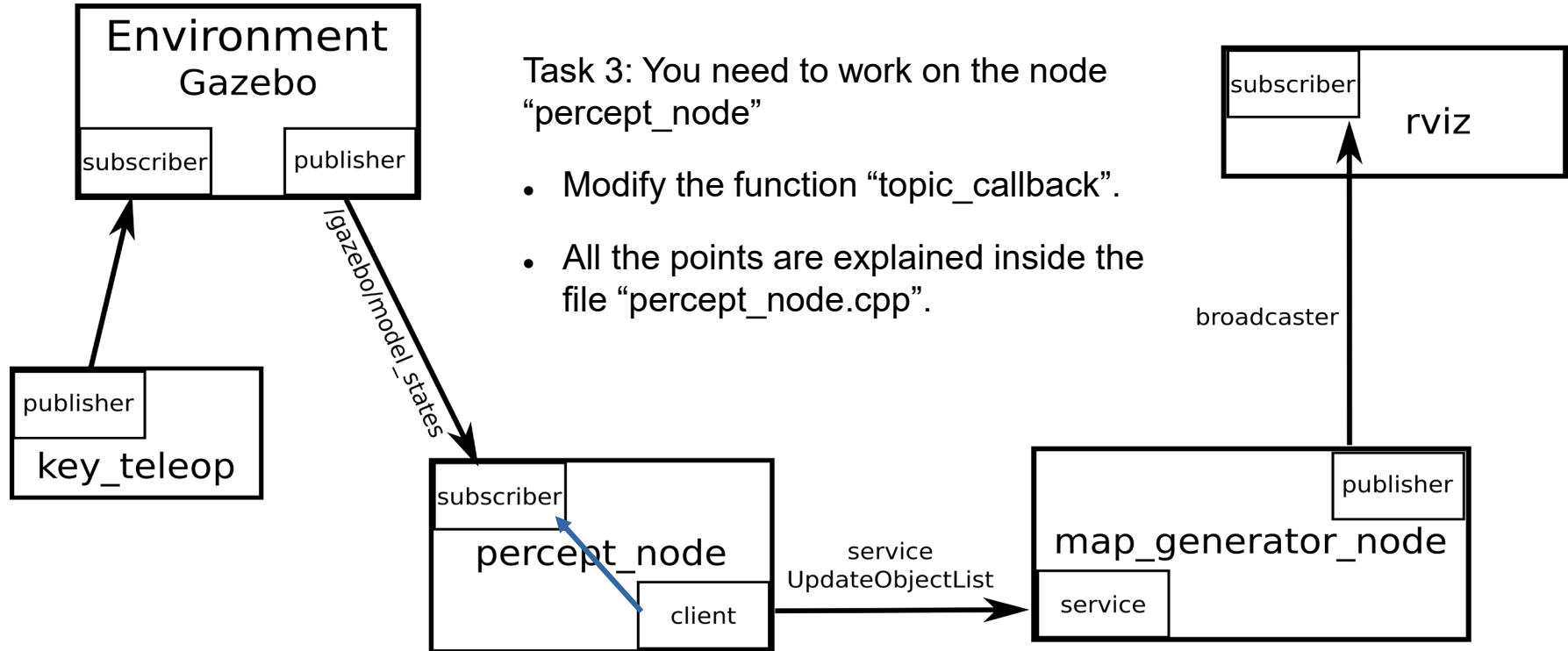
Assignment 02 – Task 3



Task 3 (4.5 points): Now you need to work on the client that is communicating with the service you created in task 2.

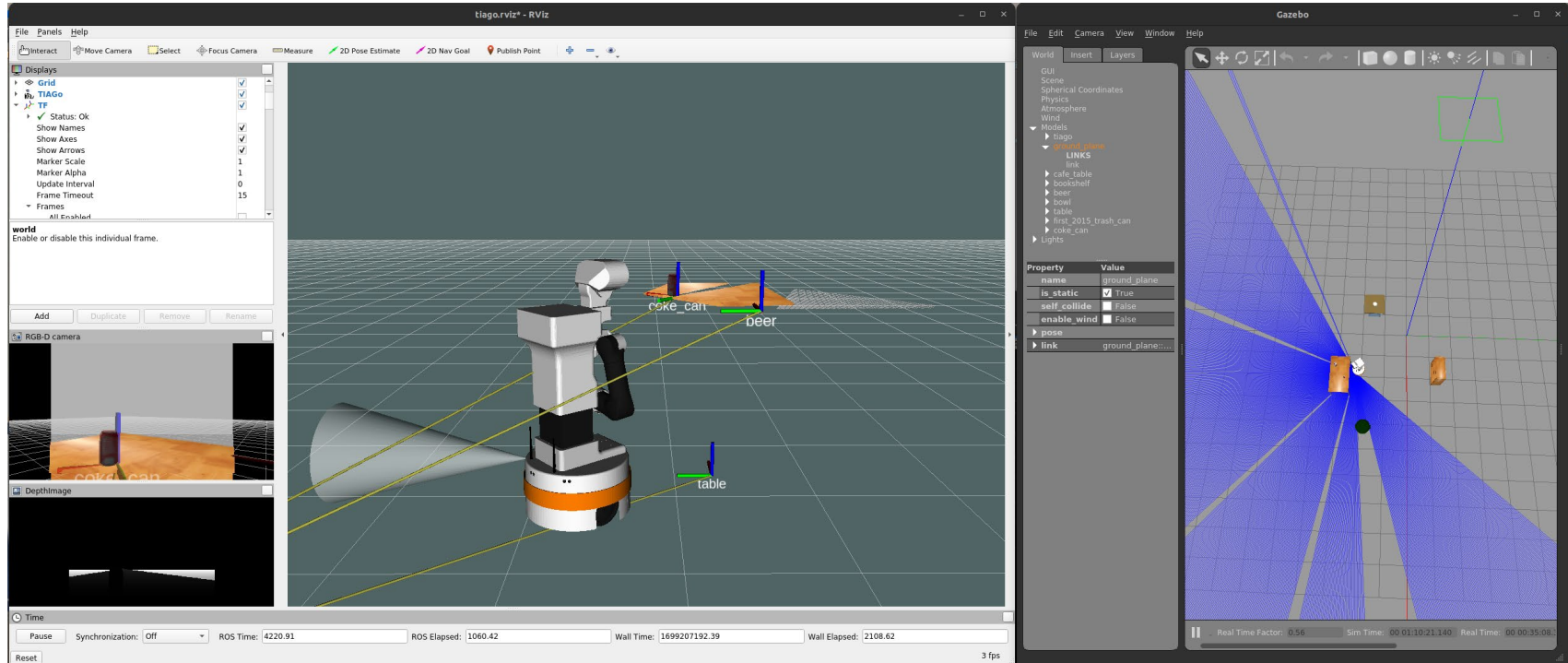
- You need to work on the file “percept_node.cpp”
- Look at the subscriber that receives information from the topic `/gazebo/model_states`
- You need to modify the function “topic_callback”. This is the logic of this callback function:
 - 1) Identify where the robot is at all times. When the robot navigates, the distance between the robot and the object(s) seen will be computed, if this distance is closer, this means that the object should be stored in a list of object names and poses.
 - 2) The robot sends the request to the service to store this information which will be published in a tf

Assignment 02 – Task 3



Assignment 02 – Task 3

Expected Outcome of Assignment 02



Assignment 02 – Task 3



Deadline for Assignment 02: **Nov 25 at 11:59 pm**

- Please upload the original material before the deadline
- Upload your whole package in a zip file
 - Within this zip file, you should include a README file that explains how to run your code. Also, include the names of your team members.
 - Only one upload per team is enough.
 - Please name the file “world_percept_assig_team_no_A02.zip”

Further reading on Prolog



Useful links:

- <https://gazebo-sim.org/docs>
- <https://wiki.ros.org/tf2/Tutorials/Writing%20a%20tf2%20broadcaster%20%28C%2B%2B%29>
- <https://wiki.ros.org/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29>



CHALMERS
UNIVERSITY OF TECHNOLOGY



CHALMERS
UNIVERSITY OF TECHNOLOGY