



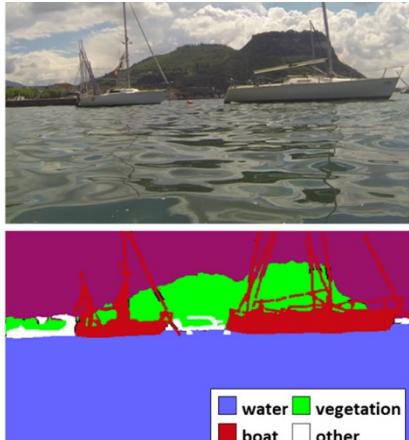
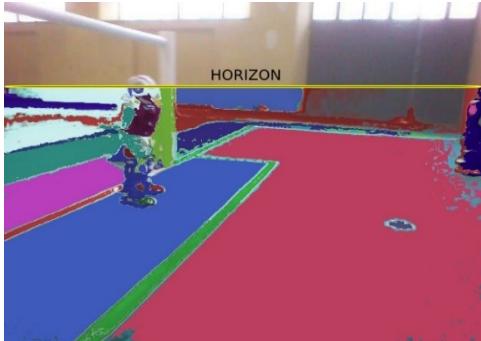
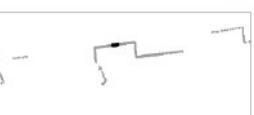
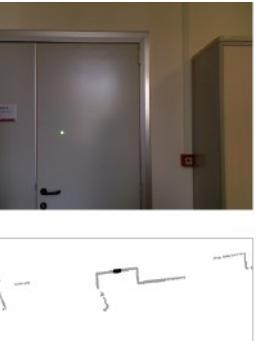
**UNIVERSITÀ DEGLI STUDI  
DELLA BASILICATA**

*Corso di Visione e Percezione*  
A.A. 2019/2020

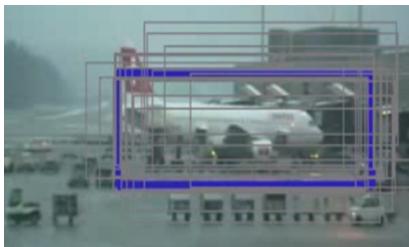
Docente  
**Domenico Daniele Bloisi**

# Simulatori

Maggio 2020



■ water ■ vegetation  
■ boat ■ other



# Il corso

---

- Home page del corso  
<http://web.unibas.it/bloisi/corsi/visione-e-percezione.html>
- Docente: Domenico Daniele Bloisi
- Periodo: Il semestre marzo 2020 – giugno 2020
- Orario:  
Martedì 17:00-19:00  
Mercoledì 8:30-10:30

# Service robots in the World

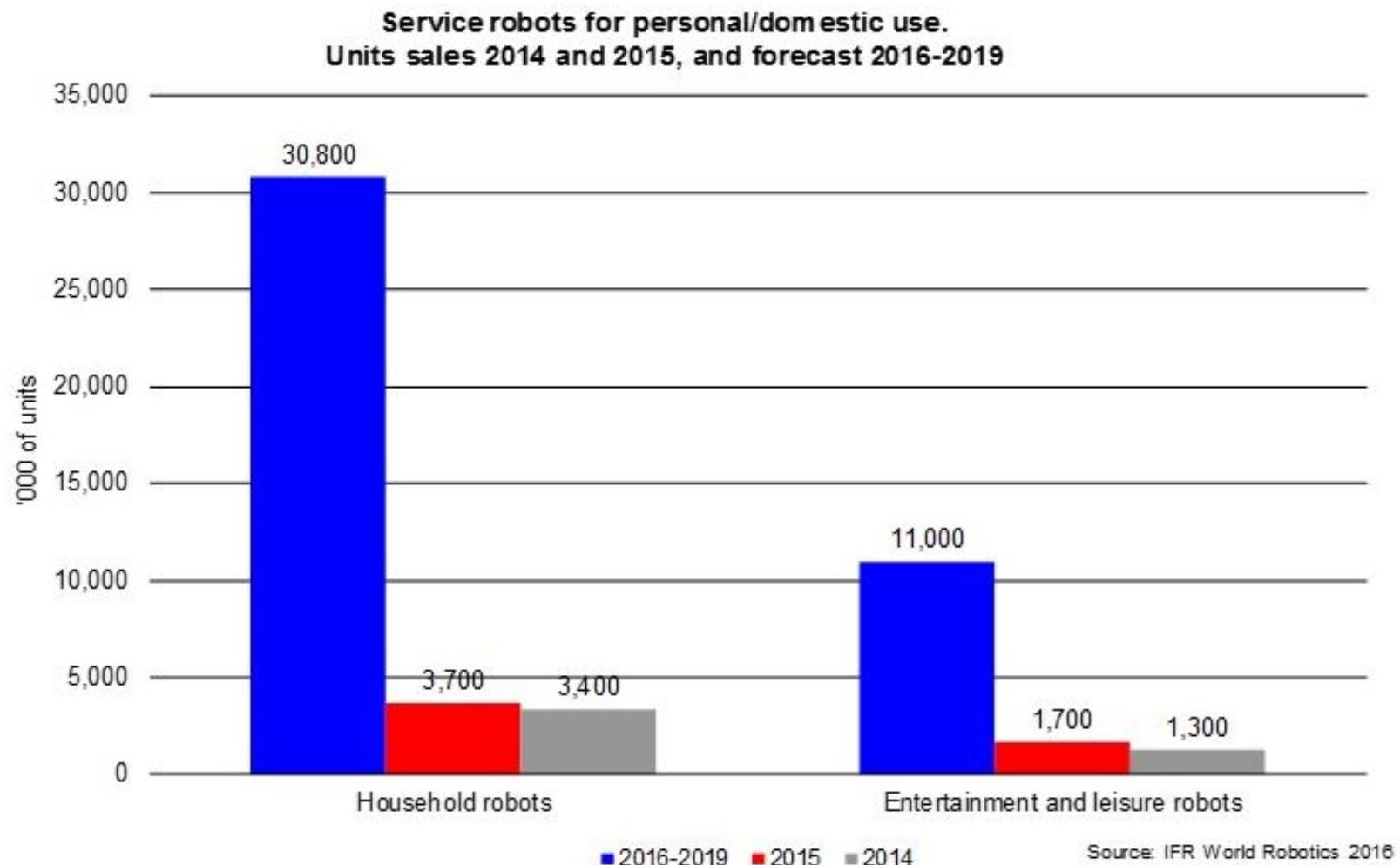
---

The worldwide number of domestic household robots will rise to 31 million between 2016 and 2019

The sales value of robots cleaning floors, mowing lawns, and cleaning swimming pools will grow to about 13 billion US dollars in this period

# Sales and forecast numbers for service robots 2014-2019

---



<https://ifr.org/ifr-press-releases/news/31-million-robots-helping-in-households-worldwide-by-2019>

# Perché usare un simulatore?

No physical dependency on the actual machine!

## Cost

- No cost for any robot or equipment
- No risk or damage, no maintenance
- No human risk

## Time

- Simulations can be run in parallel
- No battery recharge

## Experiments

- Any environment, any robot, any sensor
- Experimental repeatability
- Scalability



*400.000\$ for a beer???*

# Scegliere il giusto simulatore

---

“The best simulator does not have to resemble reality in the most accurate way. The power of a simulator is to fit to our needs.” (Elron, 1983)

## **What are we simulating?**

behavior-based, multi-robot,motion, interaction, manipulation,...

## **How are we simulating?**

rendering (3D, 2D, console), physics, ...

## **Do we need to migrate to real platforms?**

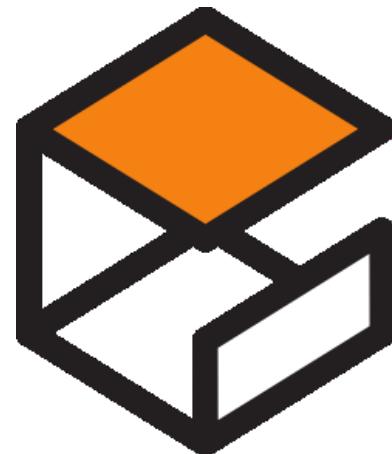
# Turtlebot 3 – simulation

---

Se non si ha a disposizione il robot reale, è possibile lavorare allo sviluppo del software del robot utilizzando un simulatore

Per il robot Tutlebot3 è possibile utilizzare  
il simulatore Gazebo

<http://gazebosim.org/>



GAZEBO

Una guida all'uso di Gazebo con il Turtlebot3 è disponibile qui

<http://emanual.robotis.com/docs/en/platform/turtlebot3/simulation>

# Installazione pacchetti per il Turtlebot3

---

Iniziamo ad installare i pacchetti ROS turtlebot3\_msgs e turtlebot3

```
$ cd ~/catkin_ws/src/  
  
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git  
  
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3.git  
  
$ cd ~/catkin_ws && catkin_make
```

[http://emanual.robotis.com/docs/en/platform/turtlebot3/pc\\_setup/#install-dependent-ros-packages](http://emanual.robotis.com/docs/en/platform/turtlebot3/pc_setup/#install-dependent-ros-packages)

# Turtlebot3 – run a simulation

---

Per poter simulare il Turtlebot 3 sul proprio PC è necessario utilizzare lo specifico ROS package **TurtleBot3 Simulations**

```
$ cd ~/catkin_ws/src  
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3\_simulations.git  
$ cd ~/catkin_ws  
$ catkin_make
```

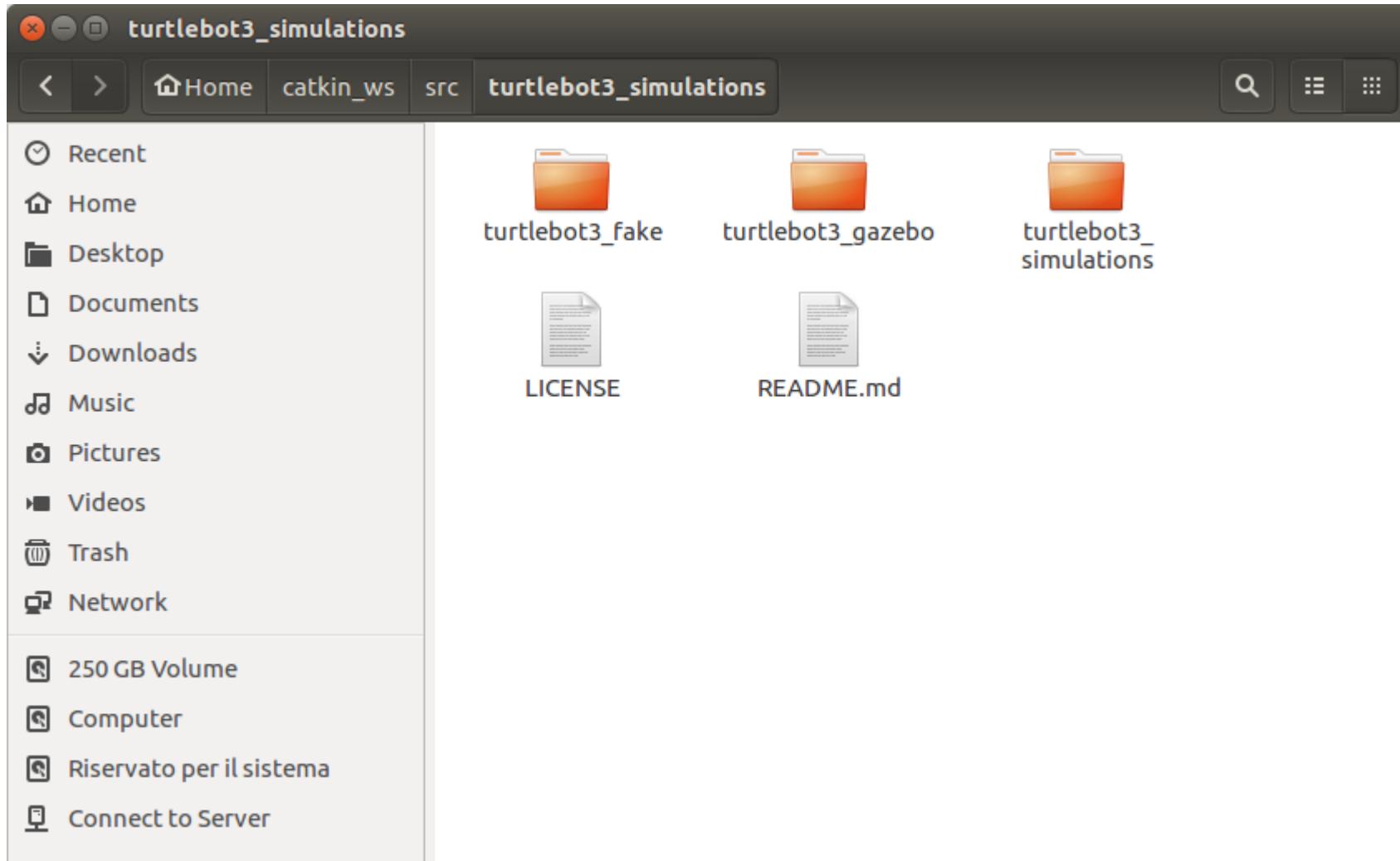
# turtlebot3\_simulations

---

```
bloisi@bloisi-U36SG: ~/catkin_ws
bloisi@bloisi-U36SG:~/catkin_ws/src$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git
Cloning into 'turtlebot3_simulations'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 1535 (delta 1), reused 1 (delta 0), pack-reused 1526
Receiving objects: 100% (1535/1535), 13.91 MiB | 2.39 MiB/s, done.
Resolving deltas: 100% (881/881), done.
Checking connectivity... done.
bloisi@bloisi-U36SG:~/catkin_ws/src$ cd ~/catkin_ws/
bloisi@bloisi-U36SG:~/catkin_ws$ catkin_make
Base path: /home/bloisi/catkin_ws
Source space: /home/bloisi/catkin_ws/src
Build space: /home/bloisi/catkin_ws/build
Devel space: /home/bloisi/catkin_ws/devel
Install space: /home/bloisi/catkin_ws/install
#####
##### Running command: "make cmake_check_build_system" in "/home/bloisi/catkin_ws/build"
#####
-- Using CATKIN_DEVEL_PREFIX: /home/bloisi/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /home/bloisi/catkin_ws/devel;/opt/ros/kinetic
-- This workspace overlays: /home/bloisi/catkin_ws/devel;/opt/ros/kinetic
-- Using PYTHON_EXECUTABLE: /usr/bin/python
-- Using Debian Python package layout
```

# turtlebot3\_simulations

---

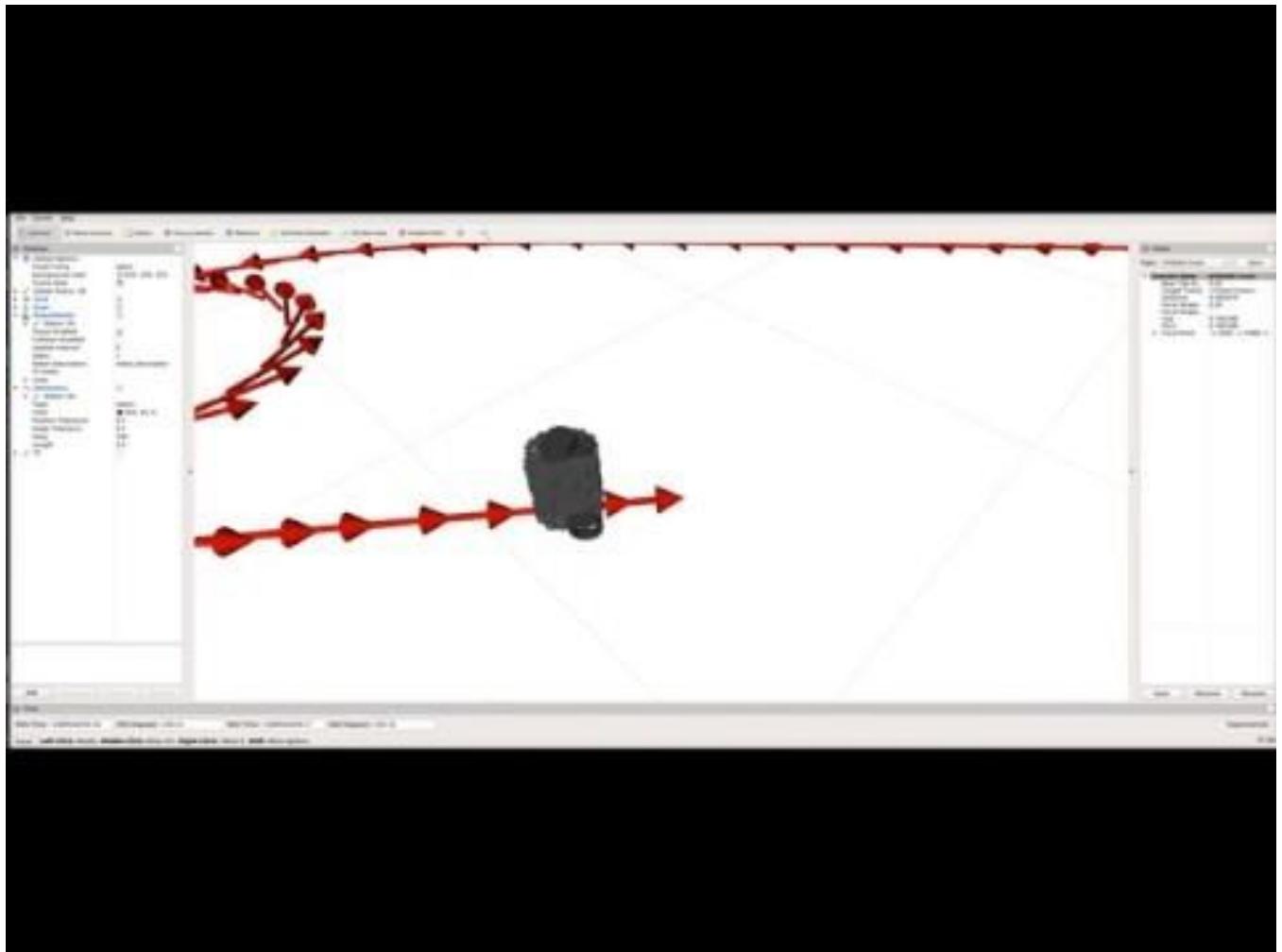


# Turtlebot3 – fake node

---

TurtleBot3 fake node è un nodo di simulazione che può essere eseguito senza necessità di avere un robot fisico.

Il TurtleBot3 virtuale può essere controllato in [RViz](#) con un teleop node.



<https://youtu.be/iHXZSLBJHMg>

# RViz

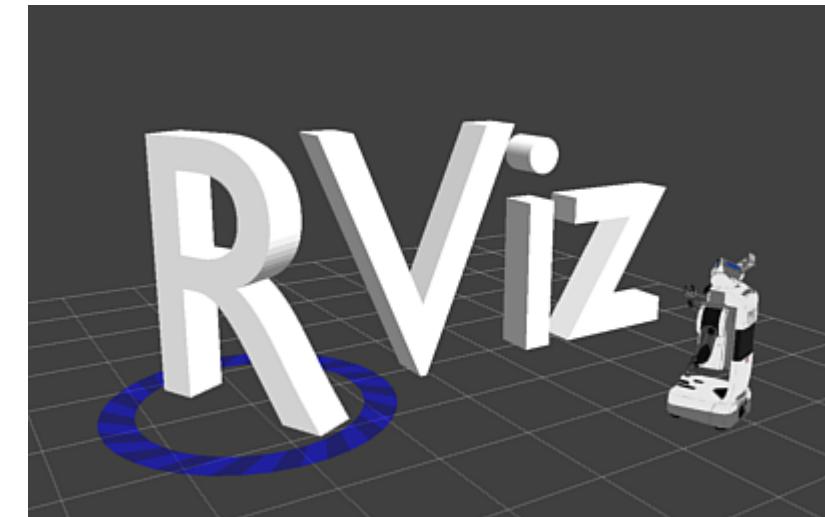
---

RViz è un tool di visualizzazione 3D di ROS che permette di percepire il mondo dalla prospettiva del robot

La documentazione relativa ad RViz è disponibile all'indirizzo  
<http://wiki.ros.org/rviz>

Per lanciare RViz è necessario eseguire  
il seguente comando

```
rosrun rviz rviz
```



# RViz

wiki.ros.org/rviz

ROSCon 2019 Diversity Scholarships: Applications Open!  
Whoever you are, whatever you do, and wherever you do it, if you're interested in ROS, then we want you to join us at ROSCon in Macau!

## ROS.org

About | Support | Discussion Forum | Service Status | Q&A answers.ros.org

Search:  Submit

Documentation    Browse Software    News    Download

### rviz

indigo    kinetic    lunar    melodic    Show EOL distros:

Documentation Status

viz: [ros\\_base](#) | [rqt\\_common\\_plugins](#) | [rqt\\_robot\\_plugins](#) | [rviz](#)

## Package Summary

✓ Released    ✓ Continuous Integration    ✓ Documented

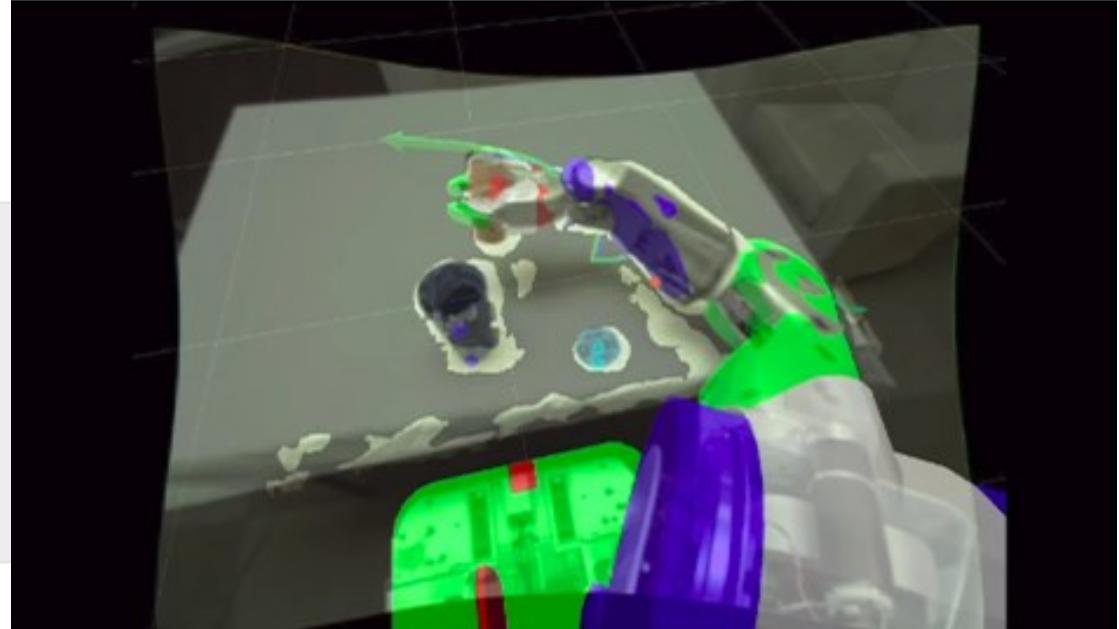
3D visualization tool for ROS.

- Maintainer status: maintained
- Maintainer: D. Hood <dhood AT osrfoundation DOT org>, William Woodall <william AT osrfoundation DOT org>
- Author: Dave Hershberger, David Gossow, Josh Faust
- License: BSD, Creative Commons
- Bug / feature tracker: <https://github.com/ros-visualization/rviz/issues>
- Source: git <https://github.com/ros-visualization/rviz.git> (branch: kinetic-devel)

Package Links

[Code API](#)  
[Tutorials](#)  
[Troubleshooting](#)  
[FAQ](#)  
[Changelog](#)  
[Change List](#)  
[Reviews](#)

Dependencies (24)  
Used by (180)  
Jenkins jobs (7)



[Contents](#)  
1 Overview

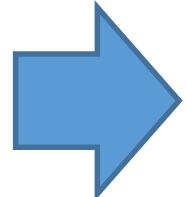
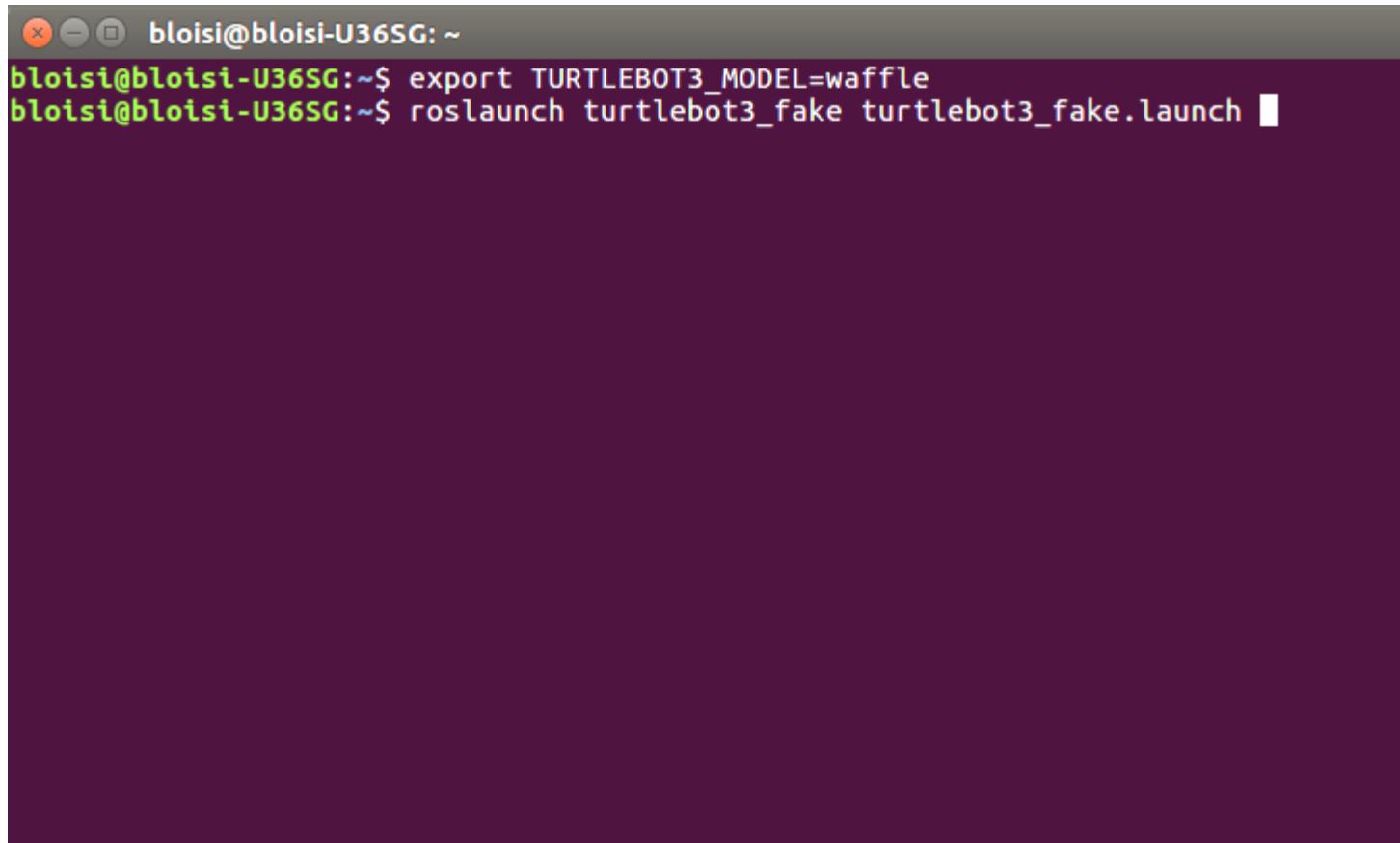
<https://youtu.be/i-Sd4xH9ZE>

# Turtlebot3 – run turtlebot3\_fake

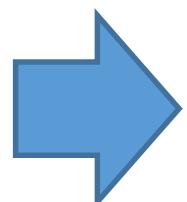
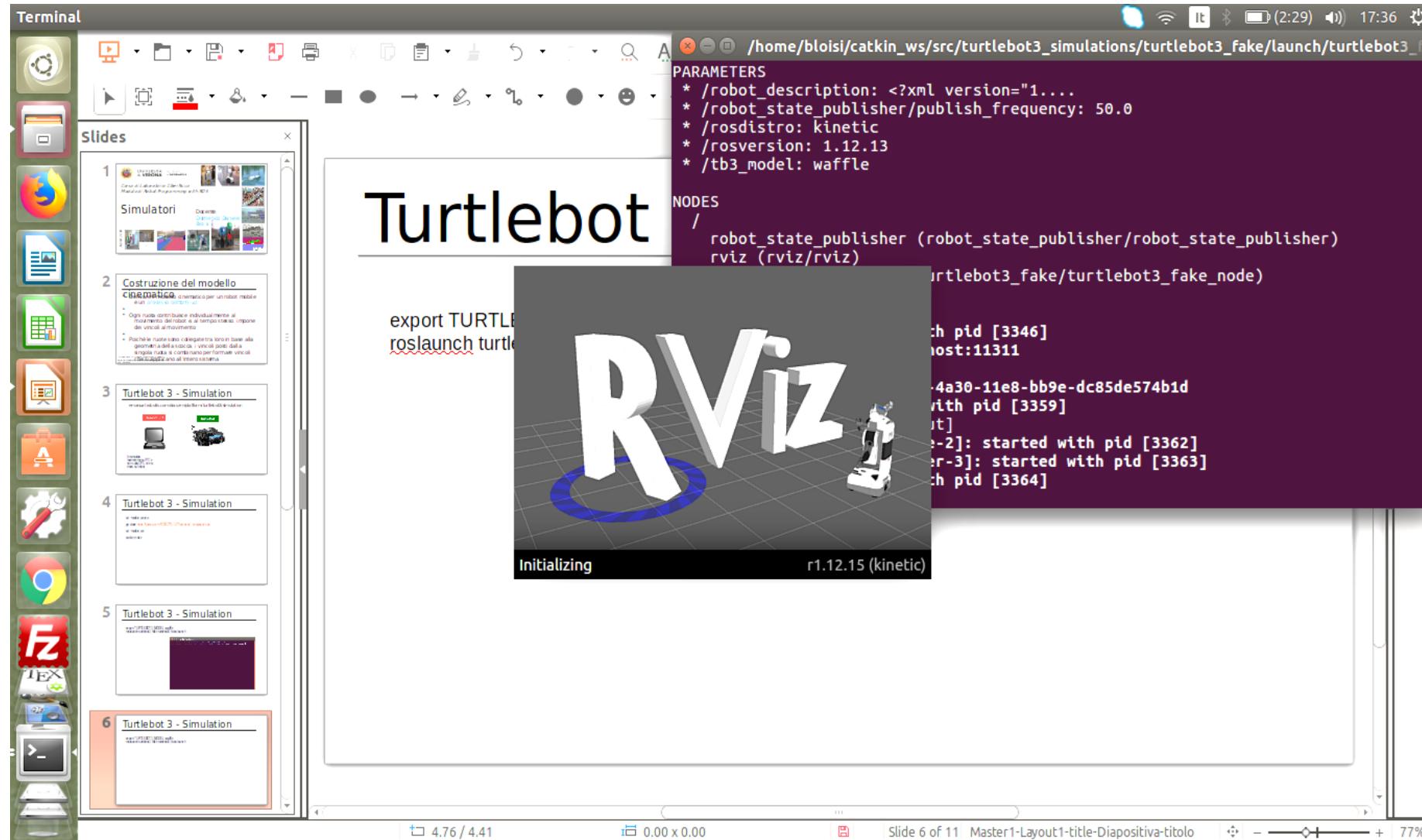
---

```
$ export TURTLEBOT3_MODEL=waffle
```

```
$ roslaunch turtlebot3_fake turtlebot3_fake.launch
```



# Turtlebot3 – run turtlebot3\_fake



# Turtlebot3 – run turtlebot3\_fake

---

```
/home/bloisi/catkin_ws/src/turtlebot3_simulations/turtlebot3_fake/launch/turtlebot3_fake.launch http://localhost:11311
bloisi@bloisi-U36SG:~/catkin_ws$ rosrun turtlebot3_fake turtlebot3_fake.launch
... logging to /home/bloisi/.ros/log/f5959c04-789d-11e9-8efc-dc85de574b1d/roslaunch-bloisi-U36SG-13348.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt

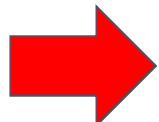
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://localhost:39600/

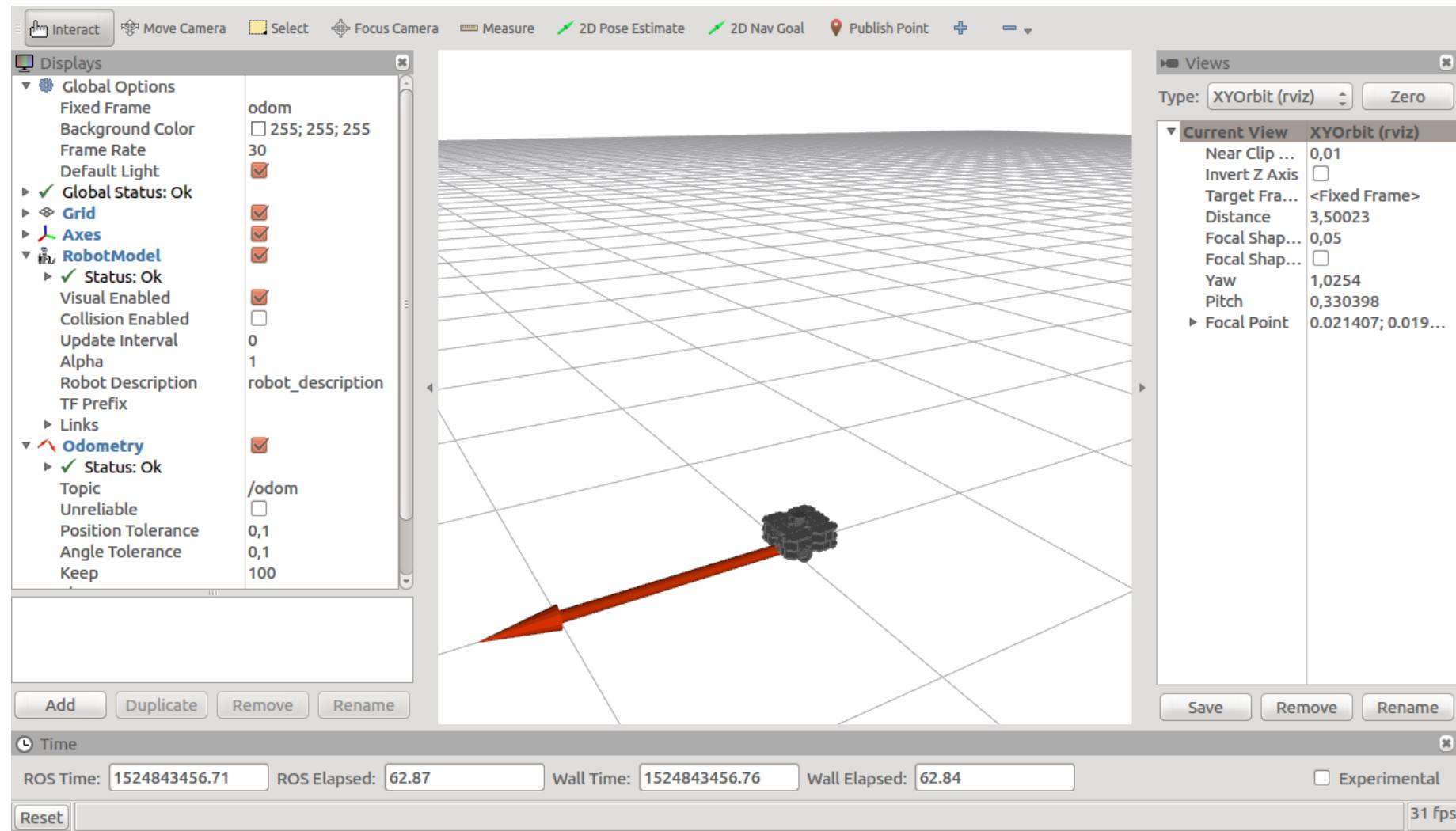
SUMMARY
=====

PARAMETERS
* /robot_description: <?xml version="1....
* /robot_state_publisher/publish_frequency: 50.0
* /rosdistro: kinetic
* /rosversion: 1.12.14
* /tb3_model: waffle

NODES
/
  robot_state_publisher (robot_state_publisher/robot_state_publisher)
  rviz (rviz/rviz)
  turtlebot3_fake_node (turtlebot3_fake/turtlebot3_fake_node)
```



# Turtlebot3 – visualizzazione



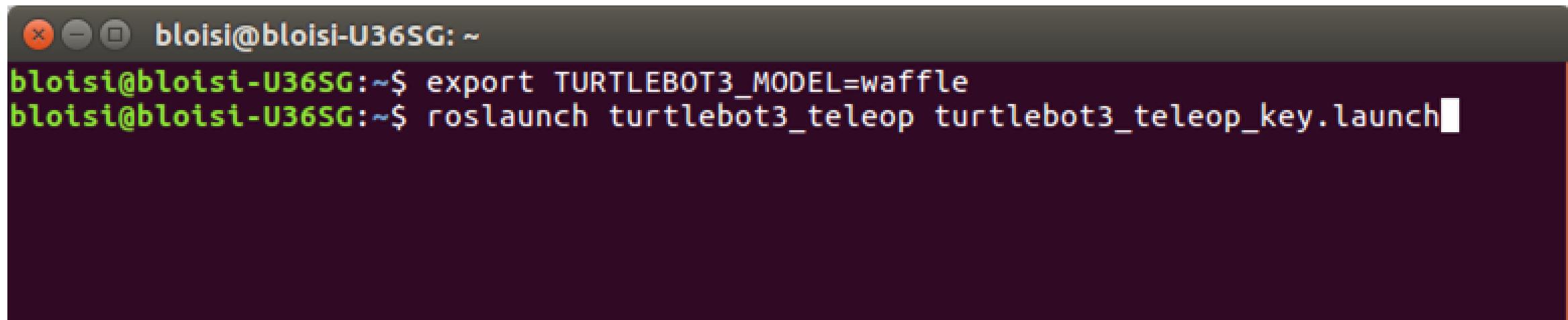
# Turtlebot3 – teleop in simulation

---

Apriamo un nuovo terminal e digitiamo

```
$ export TURTLEBOT3_MODEL=waffle
```

```
$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```



A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window contains two commands: "export TURTLEBOT3\_MODEL=waffle" and "roslaunch turtlebot3\_teleop turtlebot3\_teleop\_key.launch". The second command is partially cut off at the right edge of the window.

```
bloisi@bloisi-U36SG:~$ export TURTLEBOT3_MODEL=waffle
bloisi@bloisi-U36SG:~$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

# Turtlebot3 – i due terminal

```
x ① /home/bloisi/catkin_ws/src/turtlebot3_simulations/turtlebot3_fake/launch/turtlebot3_fake.launch http://localhost:11311
bloisi@bloisi-U36SG:~/catkin_ws$ rosrun turtlebot3_fake turtlebot3_fake.launch
... logging to /home/bloisi/.ros/log/dd735886-789e-11e9-8efc-dc85de574b1d/rosrun-bloisi-U36SG-13652.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://localhost:46414/
SUMMARY
=====
PARAMETERS
  * /robot_description: <?xml version="1....
  * /robot_state_publisher/publish_frequency: 50.0
  * /rosdistro: kinetic
  * /rosversion: 1.12.14
  * /tb3_model: waffle

NODES
  /
    robot_state_publisher (robot_state_publisher/robot_state_publisher)
    rviz (rviz/rviz)
    turtlebot3_fake_node (turtlebot3_fake/turtlebot3_fake_node)

auto-starting new master
process[master]: started with pid [13665]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to dd735886-789e-11e9-8efc-dc85de574b1d
process[rosout-1]: started with pid [13678]
started core service [/rosout]
process[turtlebot3_fake_node-2]: started with pid [13682]
process[robot_state_publisher-3]: started with pid [13687]
process[rviz-4]: started with pid [13699]
```

```
x ② /home/bloisi/catkin_ws/src/turtlebot3/teleop/launch/turtlebot3_teleop_key.launch
bloisi@bloisi-U36SG:~/catkin_ws$ rosrun turtlebot3_teleop turtlebot3_teleop_key.launch
... logging to /home/bloisi/.ros/log/dd735886-789e-11e9-8efc-dc85de574b1d/rosrun-bloisi-U36SG-13923.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://localhost:40856/
SUMMARY
=====
PARAMETERS
  * /model: waffle
  * /rosdistro: kinetic
  * /rosversion: 1.12.14

NODES
  /
    turtlebot3_teleop_keyboard (turtlebot3_teleop/turtlebot3_teleop_key)

ROS_MASTER_URI=http://localhost:11311

process[turtlebot3_teleop_keyboard-1]: started with pid [13940]

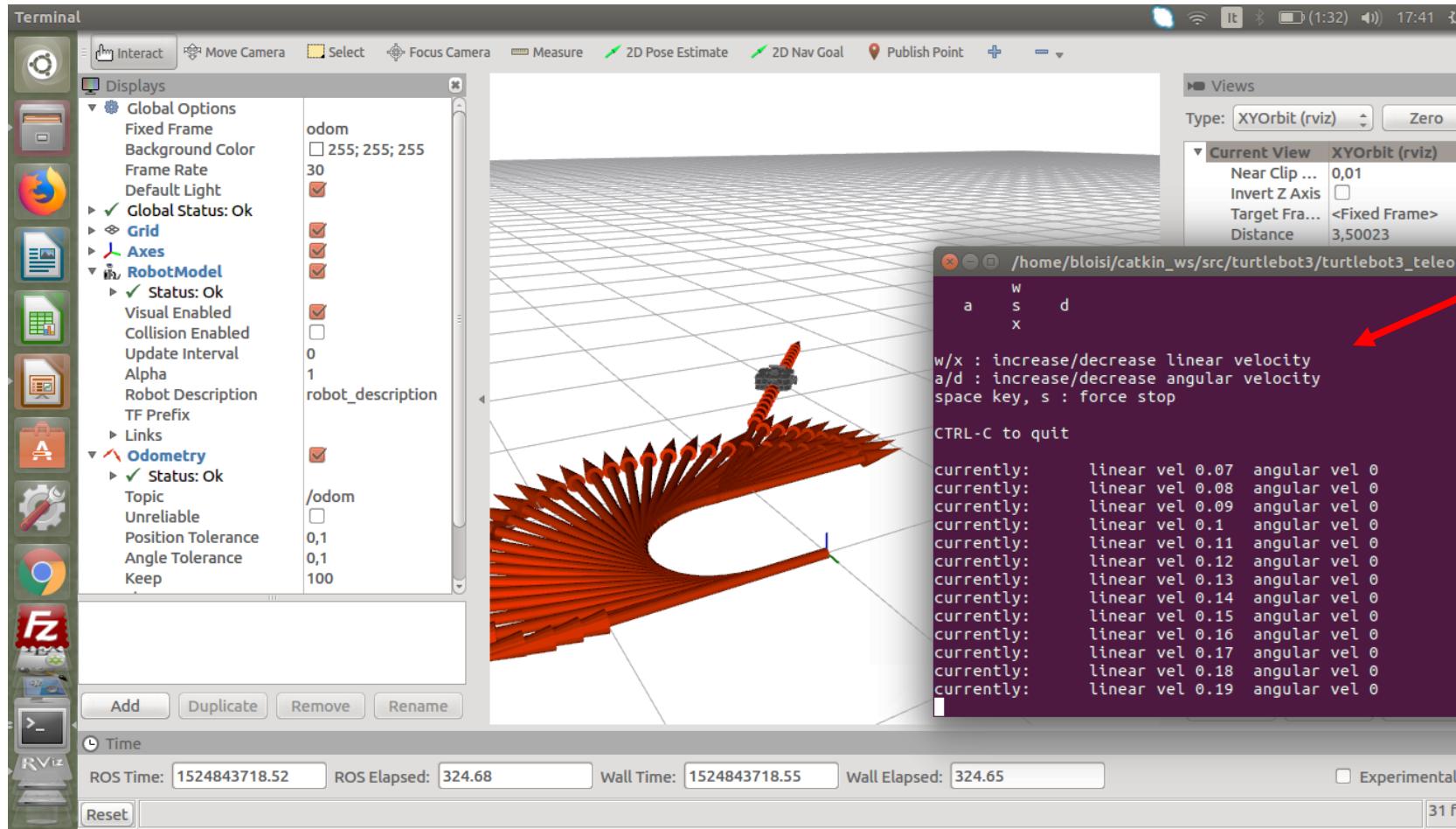
Control Your TurtleBot3!
-----
Moving around:
      w
      a   s   d
          x

w/x : increase/decrease linear velocity (Burger : ~ 0.22, Waffle and Waffle Pi : ~ 0.26)
a/d : increase/decrease angular velocity (Burger : ~ 2.84, Waffle and Waffle Pi : ~ 1.82)

space key, s : force stop

CTRL-C to quit
```

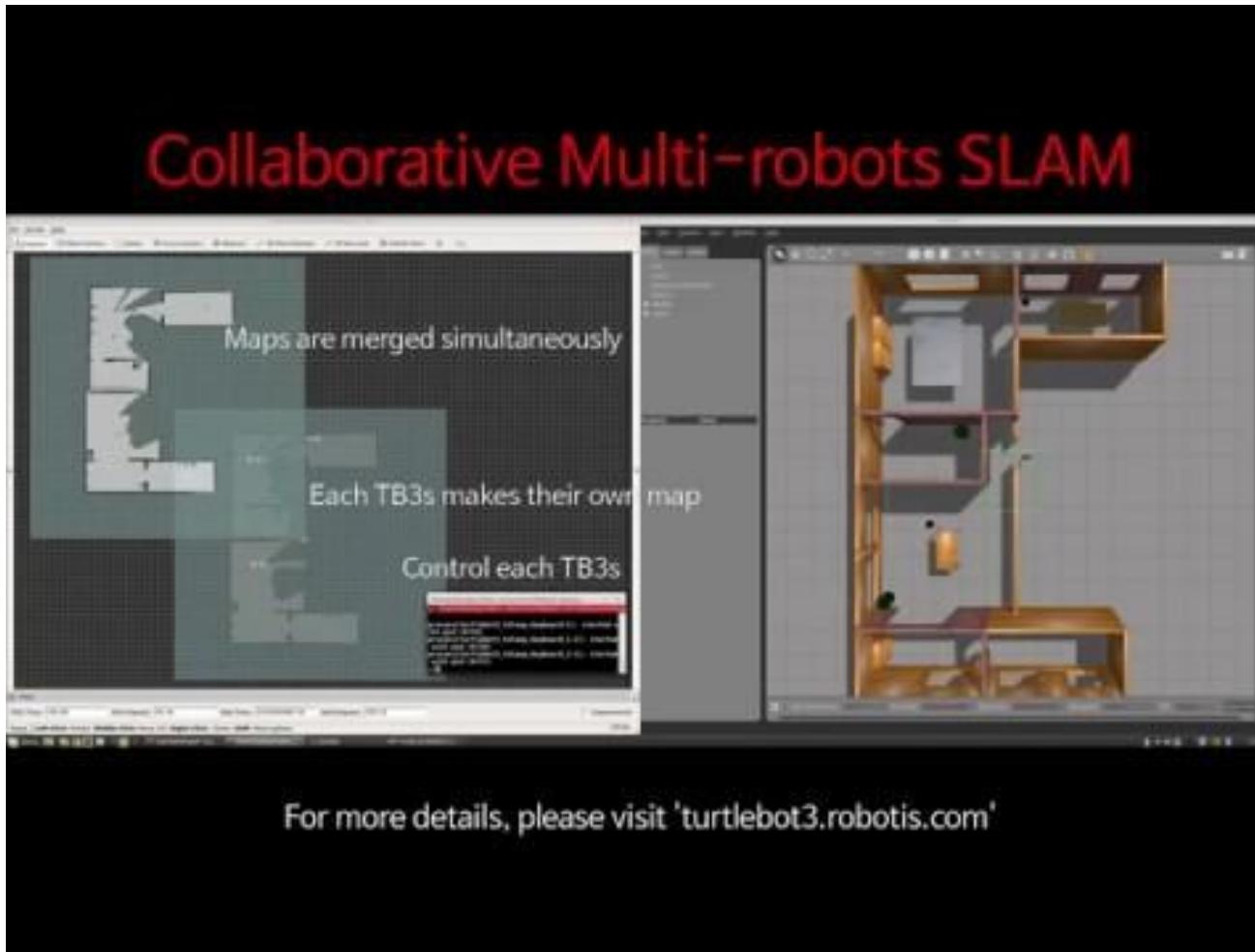
# Turtlebot3 – teleop in simulation



Per poter controllare il robot da tastiera, il terminal con il nodo teleop deve essere selezionato

# Turtlebot3 – Gazebo

---



[https://youtu.be/UzOoJ6a\\_mOg](https://youtu.be/UzOoJ6a_mOg)

# Gazebo in ROS

The screenshot shows the ROS.org website with the URL [wiki.ros.org/gazebo](http://wiki.ros.org/gazebo) in the address bar. A blue banner at the top reads "ROSCon 2019 Diversity Scholarships: Applications Open! Whoever you are, whatever you do, and wherever you do it, if you're interested in ROS, then we want you to join us at ROSCon in Macau!" Below the banner, the ROS.org logo is on the left, followed by links to "About | Support | Discussion Forum | Service Status | Q&A answers.ros.org". On the right is a search bar with a "Submit" button. The main navigation menu includes "Documentation", "Browse Software", "News", and "Download". The page title is "gazebo". It says "Only released in EOL distros: electric fuerte groovy". Under "Documentation Status", there's a link to "simulator\_gazebo". The "Package Summary" section includes a "Documented" badge, a description of the package, and a list of contributors. To the right, there's a sidebar with "Package Links" (Code API, Msg/Srv API, gazebo website, Tutorials, Troubleshooting, FAQ, Change List, Reviews), "Dependencies (26)", "Used by (45)", and a "More Actions" dropdown. The sidebar also has sections for "Wiki" (Distributions, ROS/Installation, ROS/Tutorials, RecentChanges), "Page" (Immutable Page, Info, Attachments), and "User" (Login).

<http://wiki.ros.org/gazebo>

# Gazebo + ROS



Meta Package: `gazebo_ros_pkgs`

<b>gazebo</b> Stand Alone Core  urdfdom	<b>gazebo_msgs</b> Msg and Srv data structures for interacting with Gazebo from ROS.	<b>gazebo_tests</b> <small>Merged to gazebo_plugins</small> Contains a variety of unit tests for gazebo, tools and plugins.	<b>gazebo_worlds</b> <small>Merged to gazebo_ros</small> Contains a variety of unit tests for gazebo, tools and plugins.  wg simple_erratic simple_office wg_collada_throttled - delete wg_collada grasp empty_throttled 3stacks elevator simple_office_table scan empty simple balcony camera test_friction simple_office2 empty_listener	<b>gazebo_tools</b> <small>Removed</small>
<b>gazebo_ros</b> Formerly <code>simulator_gazebo/gazebo</code>  This package wraps <code>gzserver</code> and <code>gzclient</code> by using two Gazebo plugins that provide the necessary ROS interface for messages, services and dynamic reconfigure  ROS node name: <code>gazebo</code>  Plugins: <code>gazebo_ros_api_plugin</code> <code>gazebo_ros_paths_plugin</code>  Usage: <code>rosrun gazebo_ros gazebo</code> <code>rosrun gazebo_ros gzserver</code> <code>rosrun gazebo_ros gzclient</code> <code>rosrun gazebo_ros spawn_model</code> <code>rosrun gazebo_ros perf</code> <code>rosrun gazebo_ros debug</code>	<b>gazebo_plugins</b> Robot-Independent Gazebo plugins.  Sensory <code>gazebo_ros_projector</code> <code>gazebo_ros_p3d</code> <code>gazebo_ros_imu</code> <code>gazebo_ros_laser</code> <code>gazebo_ros_f3d</code> <code>gazebo_ros_camera_utils</code> <code>gazebo_ros_depth_camera</code> <code>gazebo_ros_openni_kinect</code> <code>gazebo_ros_camera</code> <code>gazebo_ros_bumper</code> <code>gazebo_ros_block_laser</code> <code>gazebo_ros_gpu_laser</code>  Motory <code>gazebo_ros_joint_trajectory</code> <code>gazebo_ros_diffdrive</code> <code>gazebo_ros_force</code> <code>gazebo_ros_template</code>  Dynamic Reconfigure <code>vision_reconfigure</code> <code>hokuyo_node</code> <code>camera_synchronizer</code>			<b>gazebo_ros_paths_plugin</b> Provides ROS package paths to Gazebo

**gazebo\_ros\_api\_plugin**

Gazebo Subscribed Topics

`~set_link_state`  
`~set_model_state`

Gazebo Published Parameters

`/use_sim_time`

Gazebo Published Topics

`/clock`  
`~/link_states`  
`~/model_states`

Gazebo Services

`~spawn_urdf_model`  
`~spawn_sdf_model`  
`~/delete_model`

State and properties getters

...

State and properties setters

...

Simulation control

`~/pause_physics`  
`~/unpause_physics`  
`~/reset_simulation`  
`~/reset_world`

Force control

`~/apply_body_wrench`  
`~/apply_joint_effort`  
`~/clear_joint_forces`  
`~/clear_body_wrenches`

ROS packages

Gazebo Plugin

Deprecated from `simulator_gazebo`

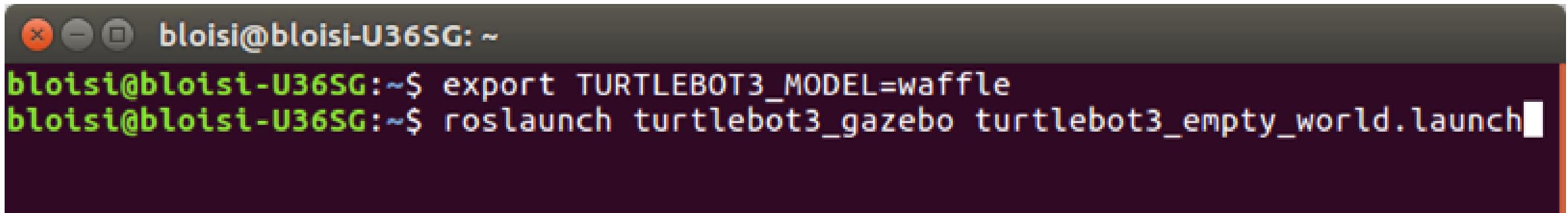
# Turtlebot3 – empty world

---

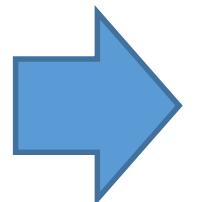
Apriamo un nuovo terminal e digitiamo

```
$ export TURTLEBOT3_MODEL=waffle
```

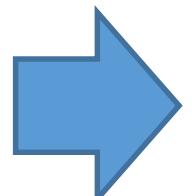
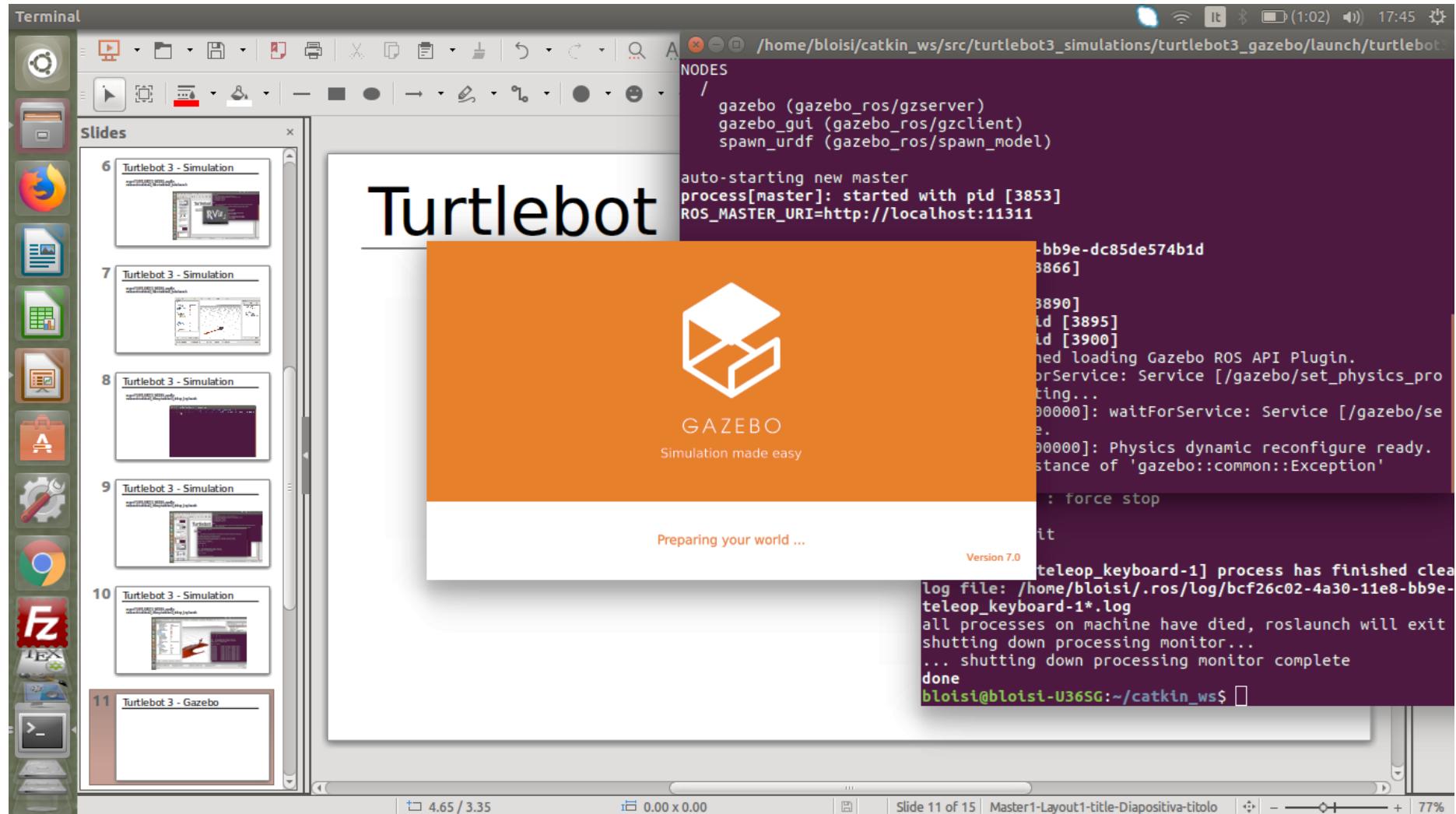
```
$ roslaunch turtlebot3_gazebo turtlebot3_empty_world.launch
```



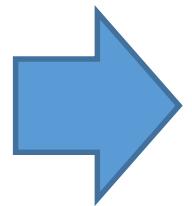
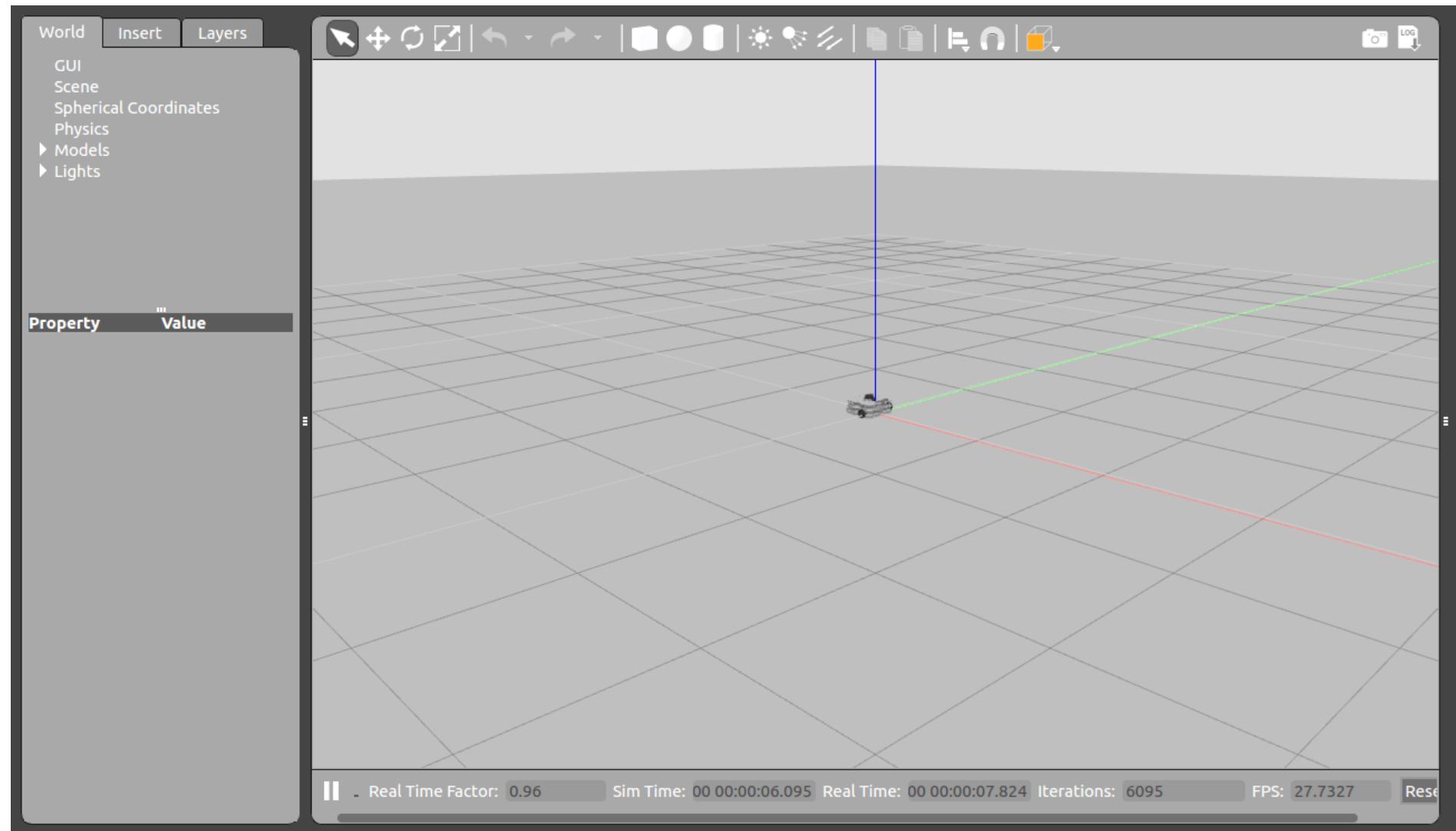
A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window contains two commands: "export TURTLEBOT3\_MODEL=waffle" in red text and "roslaunch turtlebot3\_gazebo turtlebot3\_empty\_world.launch" in green text. The terminal has a dark background with light-colored text.



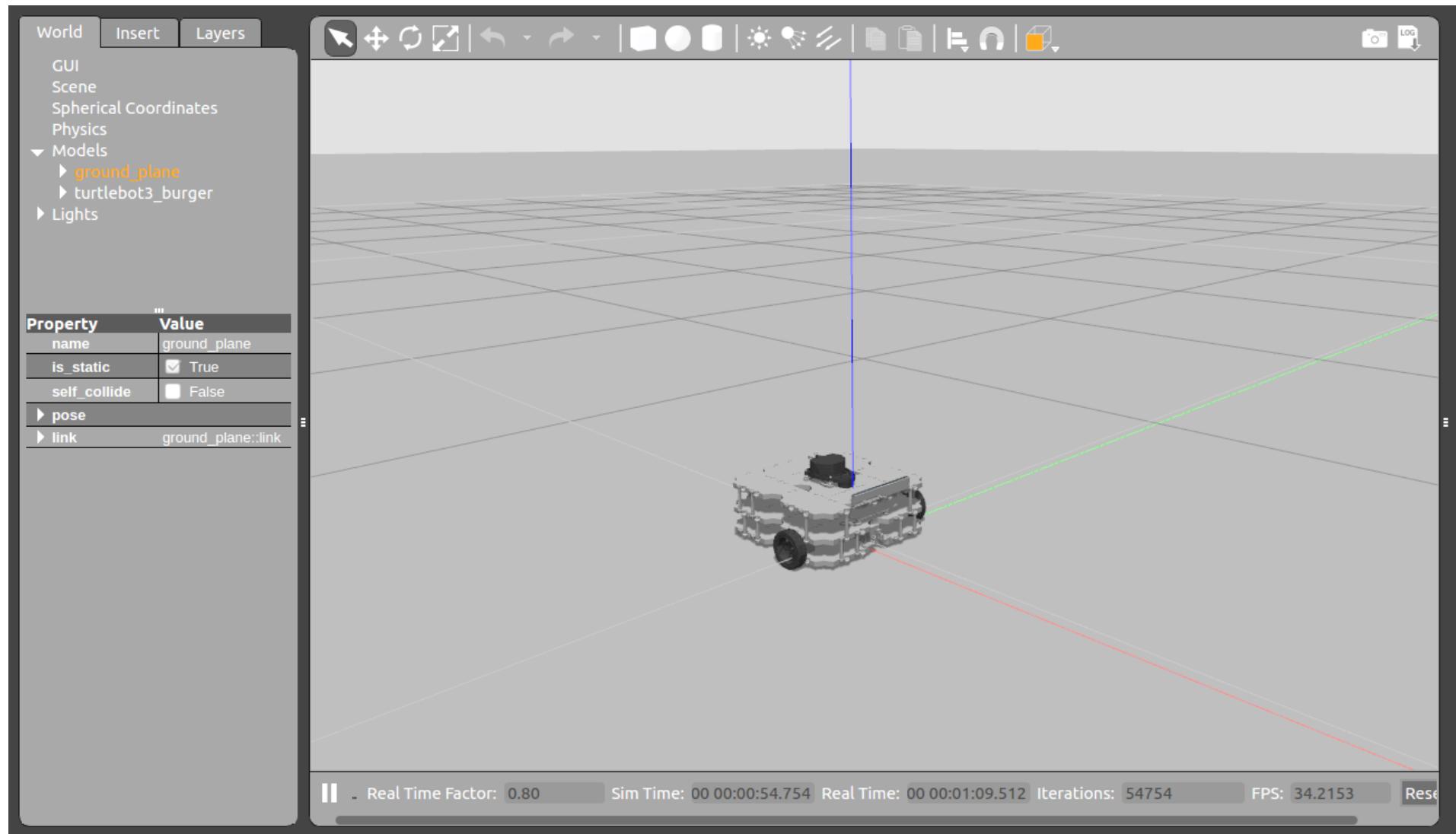
# Turtlebot3 – empty world



# Turtlebot3 – empty world



# Turtlebot3 – empty world



# Turtlebot3 – Turtlebot3 World

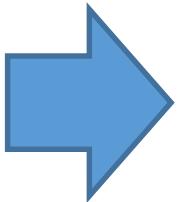
---

```
$ export TURTLEBOT3_MODEL=waffle
```

```
$ roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

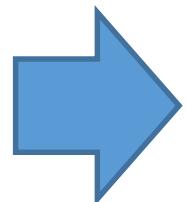
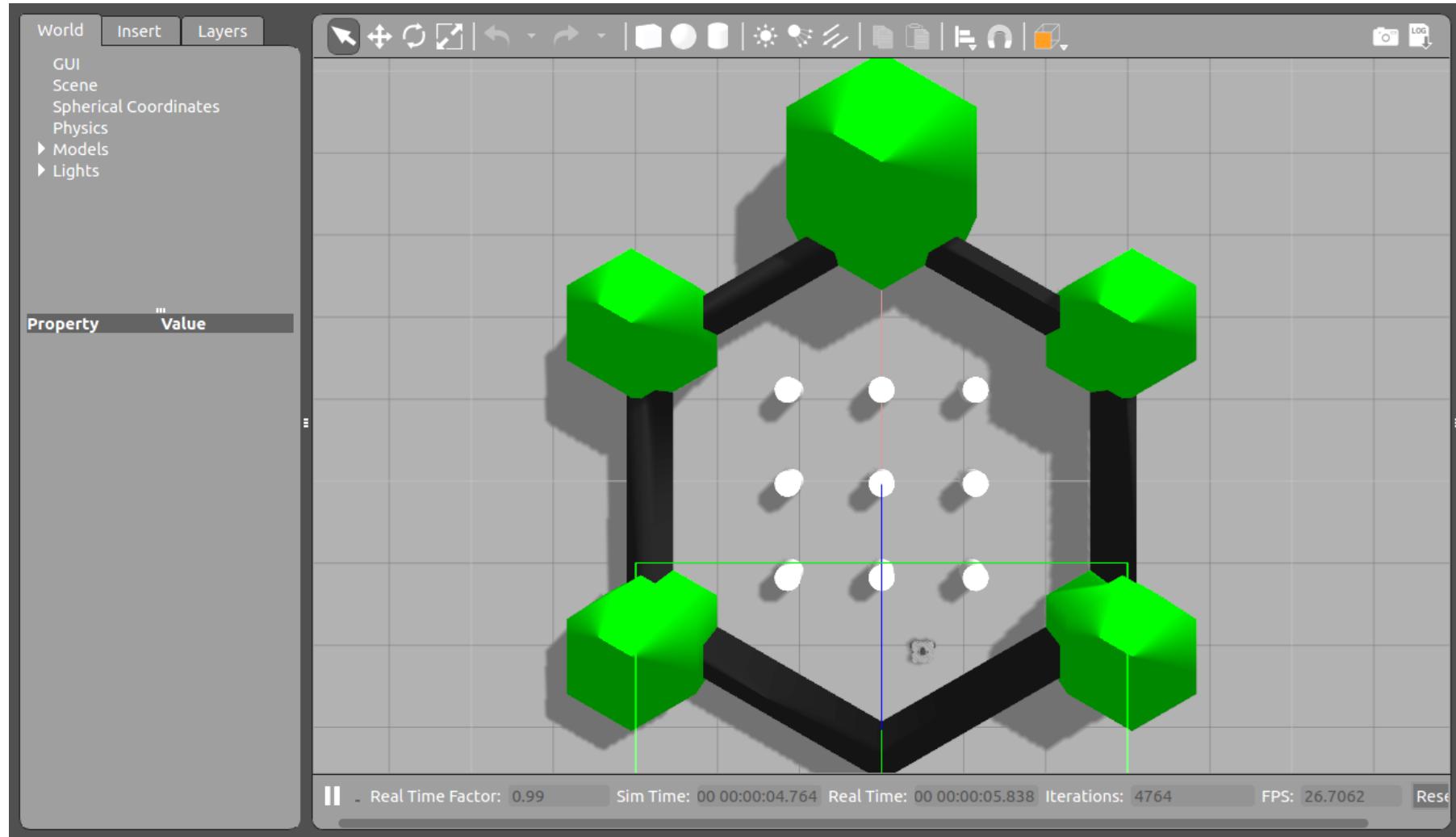
A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". It contains two commands: "export TURTLEBOT3\_MODEL=waffle" and "roslaunch turtlebot3\_gazebo turtlebot3\_world.launch".

```
bloisi@bloisi-U36SG:~$ export TURTLEBOT3_MODEL=waffle
bloisi@bloisi-U36SG:~$ roslaunch turtlebot3_gazebo turtlebot3_world.launch
```



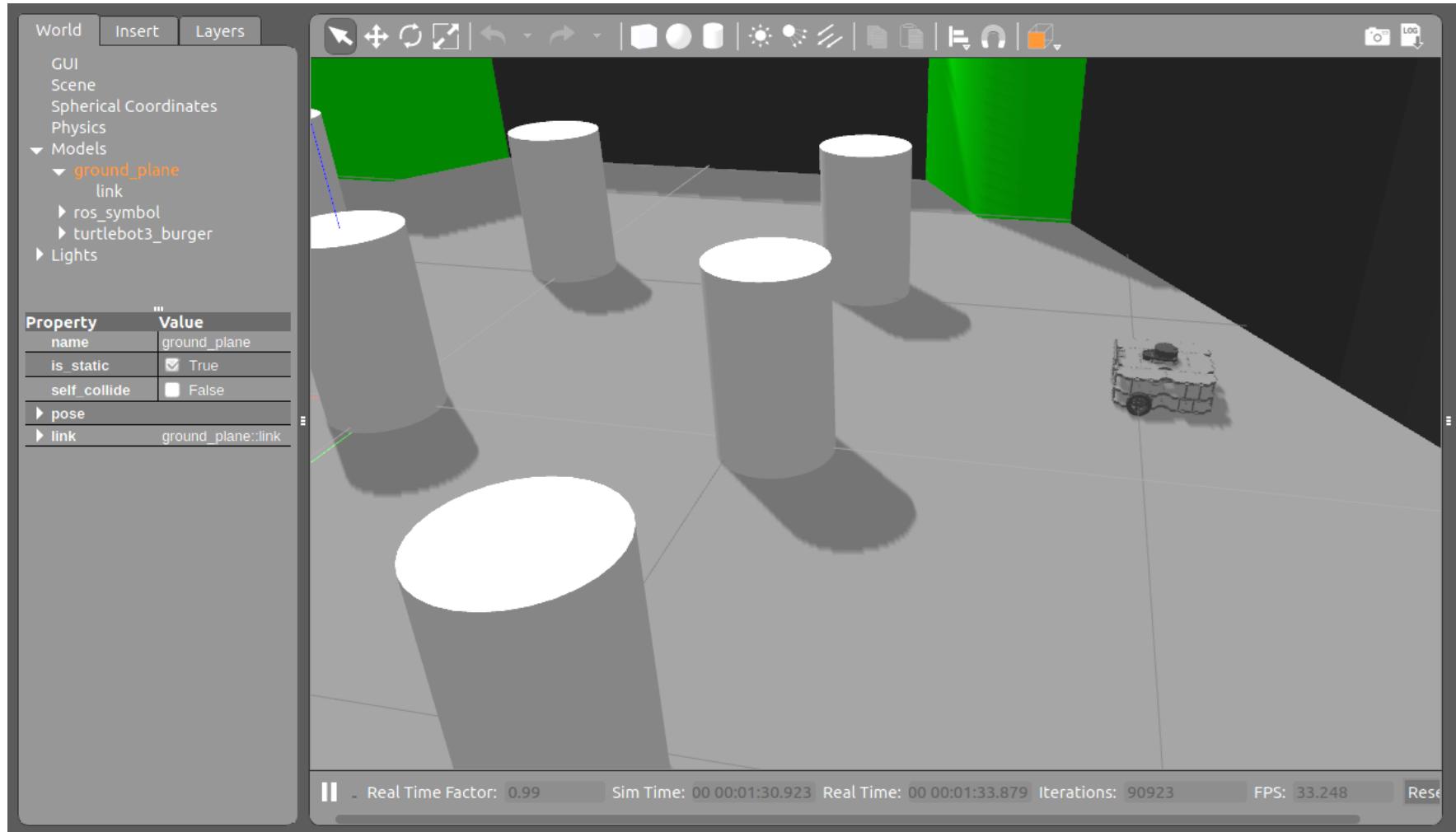
# Turtlebot3 – Turtlebot3 World

---

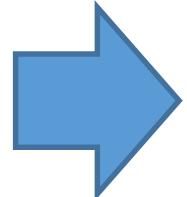
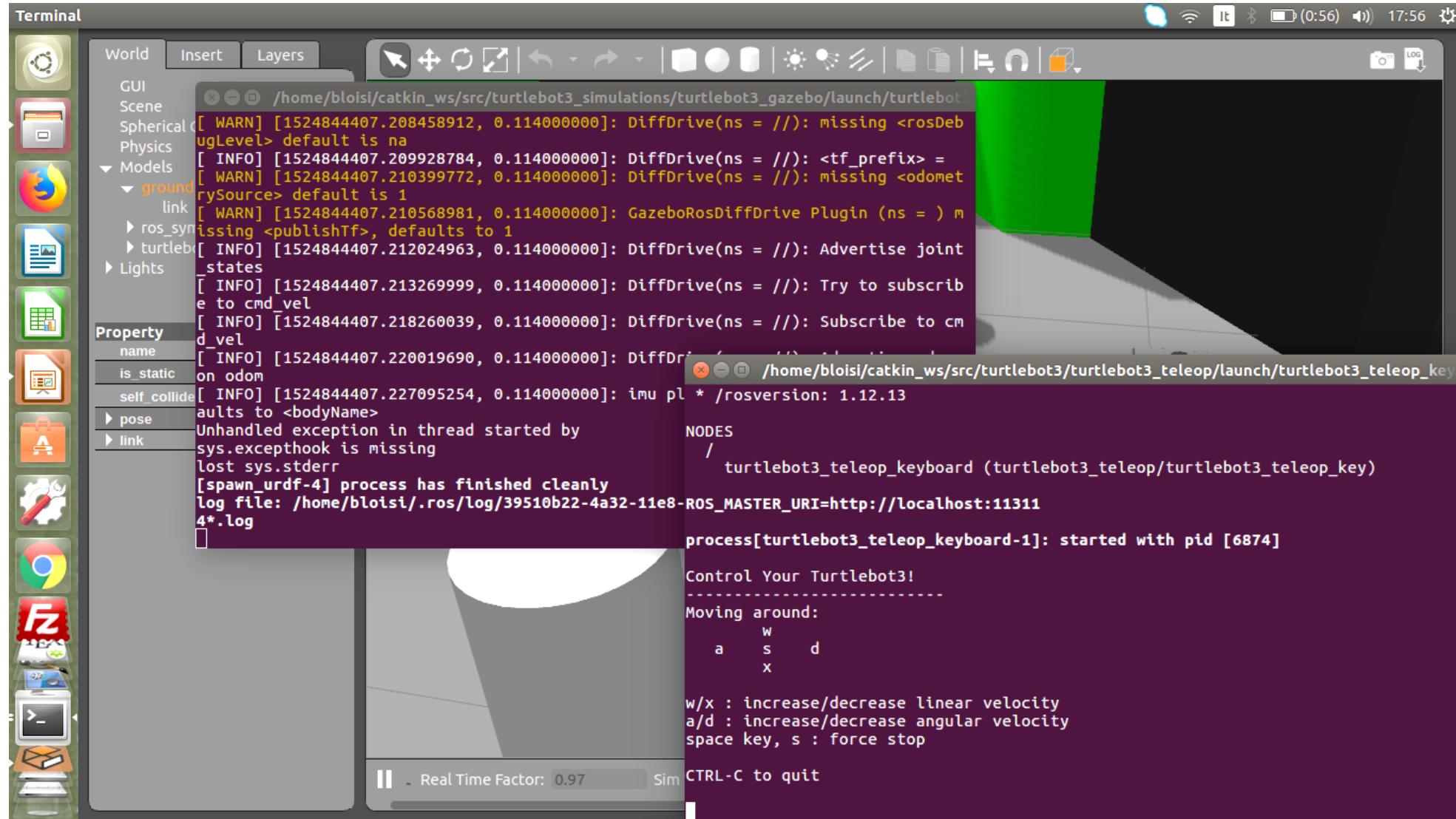


# Turtlebot3 – Turtlebot3 World

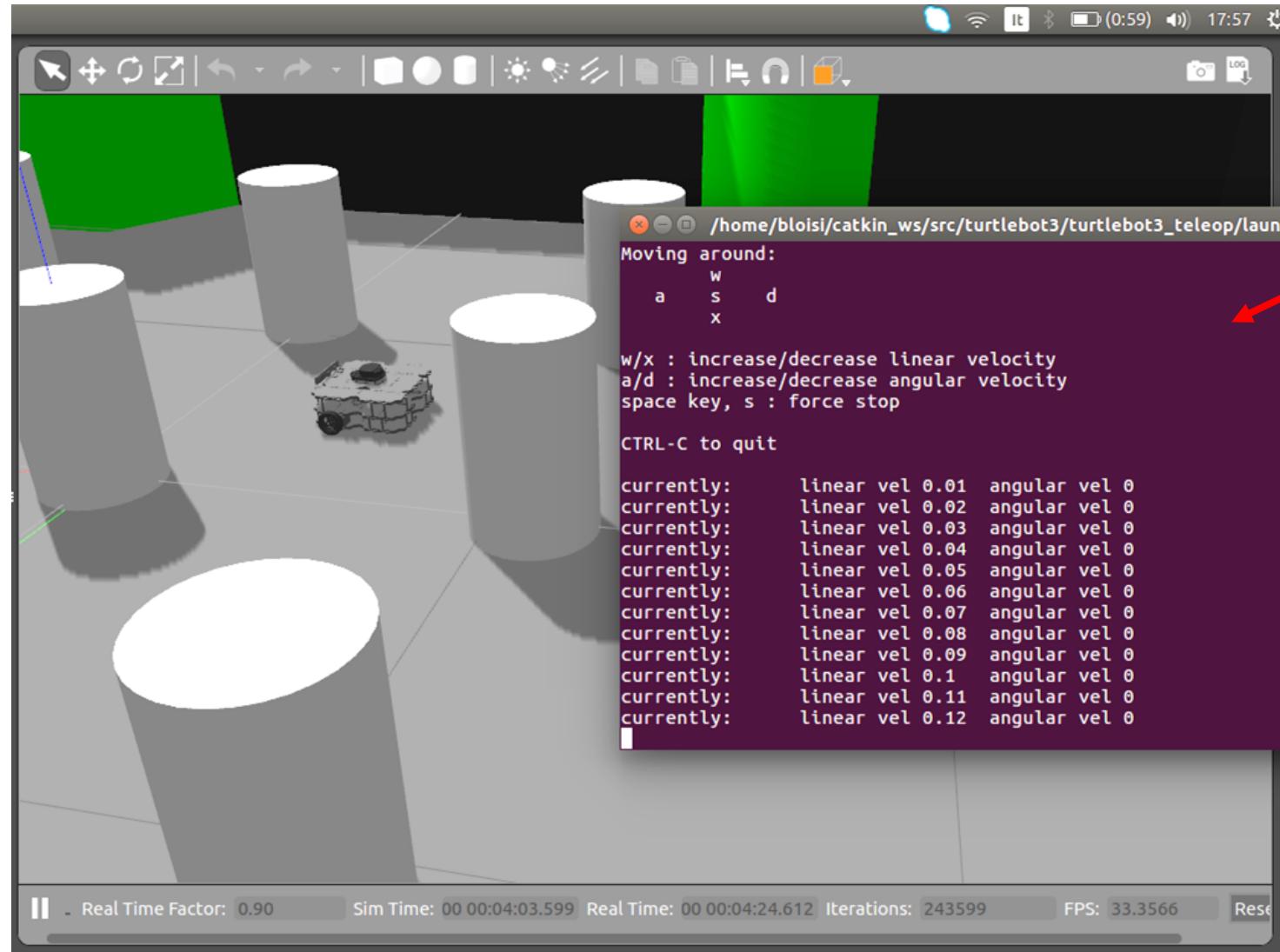
---



# Teleoperation in Turtlebot3 World



# Teleoperation in Turtlebot3 World

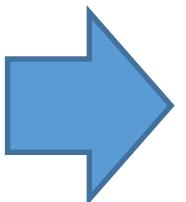
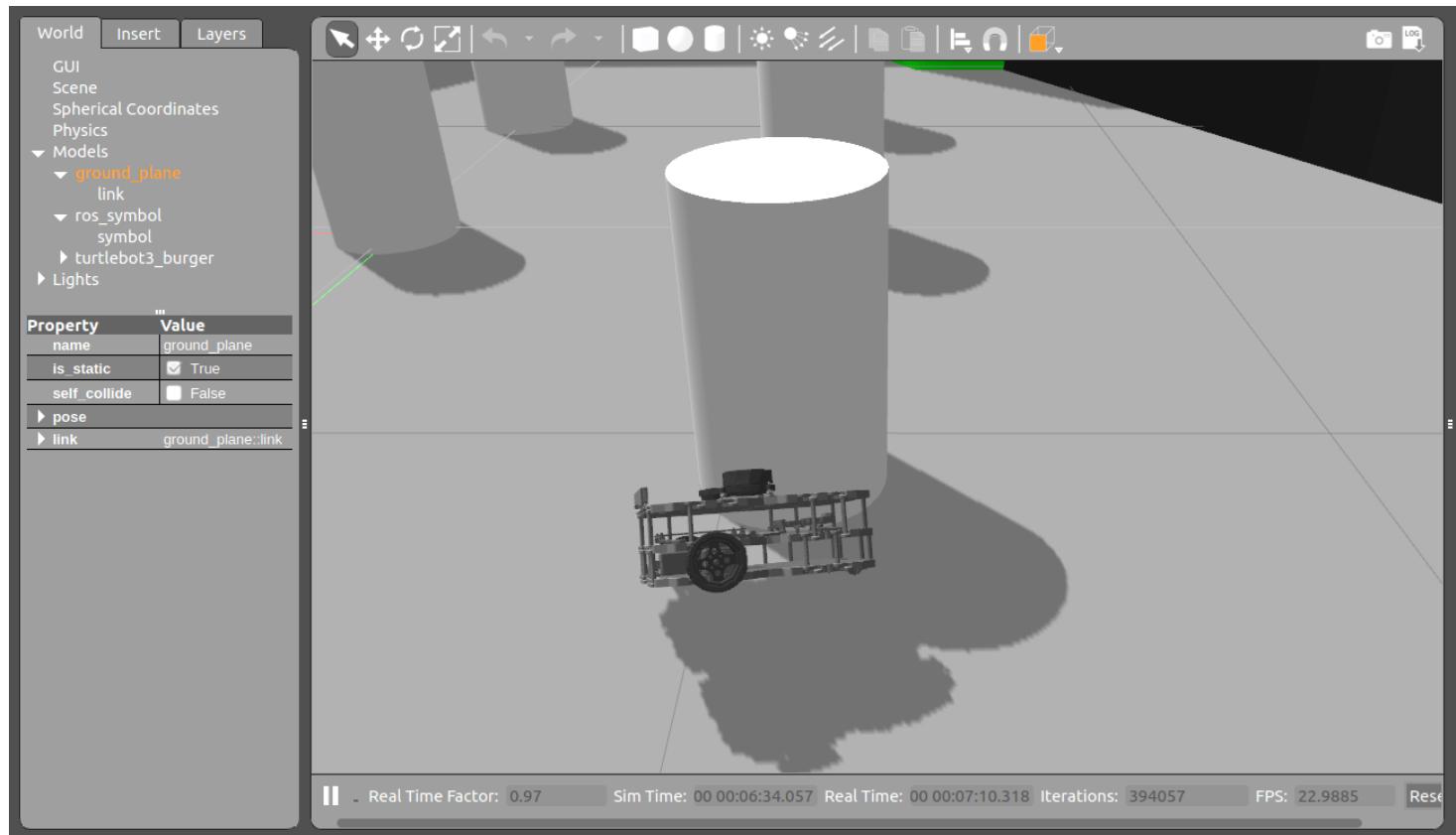


Per poter controllare il robot da tastiera, il terminal con il nodo teleop deve essere selezionato

# Esercizio Turtlebot3 World

---

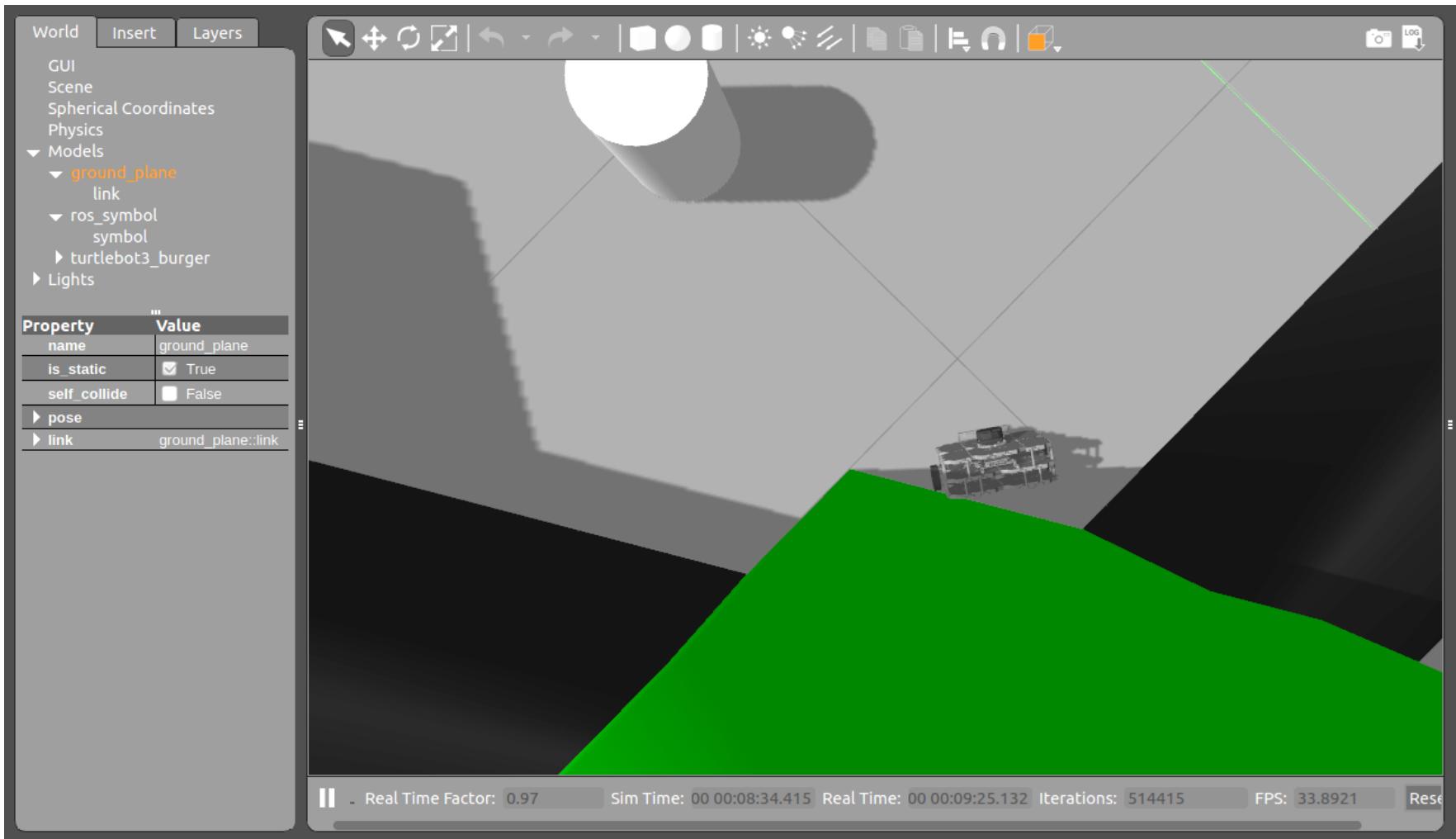
Utilizzando il nodo di teleoperazione, provare a posizionare il robot su una sola ruota



# Esercizio Turtlebot3 World

---

## Esempio



# Turtlebot3 – collision avoidance

---

## Terminale 1

Lanciare il nodo per la simulazione del Turtlebot3 World

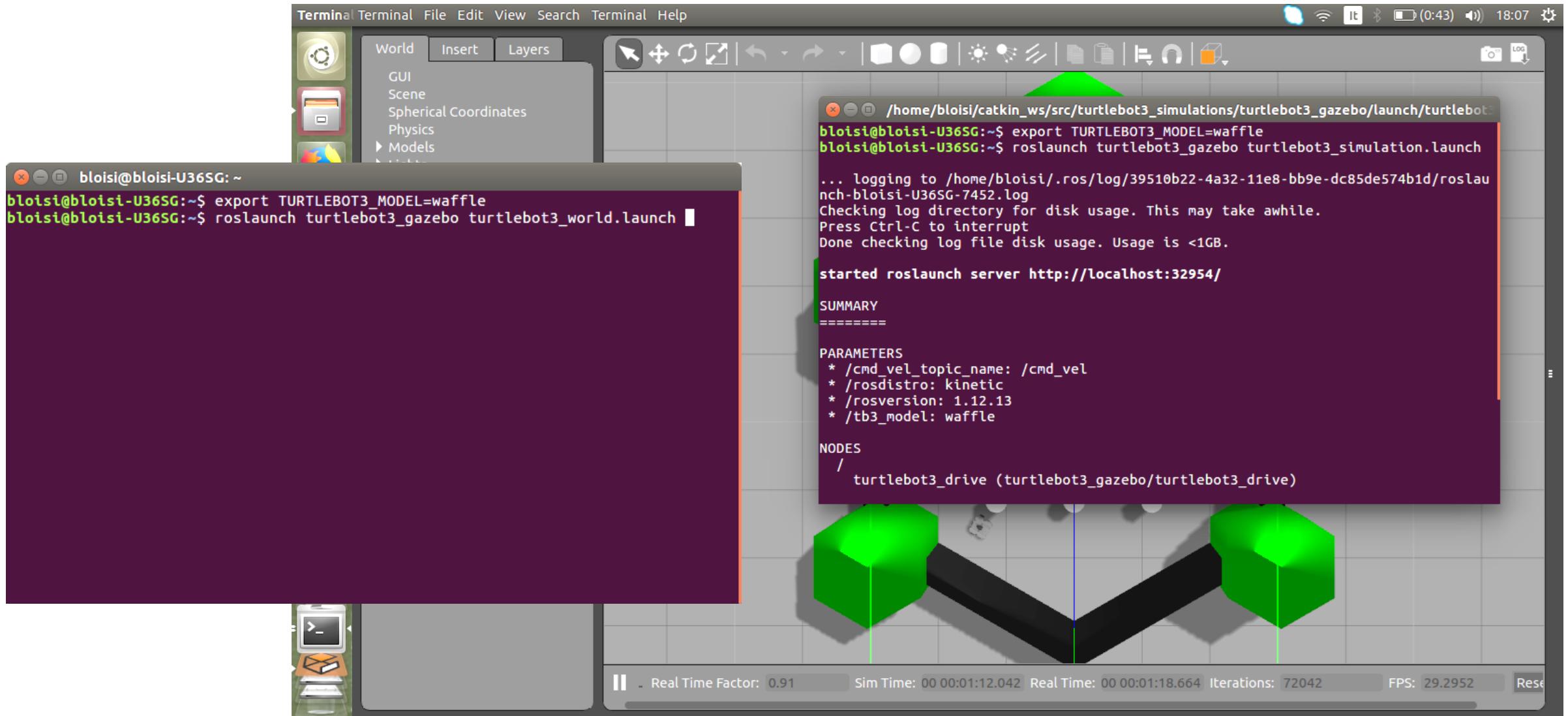
```
$ export TURTLEBOT3_MODEL=waffle  
$ roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

## Terminale 2

Lanciare il nodo per l'autonomous drive

```
$ export TURTLEBOT3_MODEL=waffle  
$ roslaunch turtlebot3_gazebo turtlebot3_simulation.launch
```

# Turtlebot3 – collision avoidance

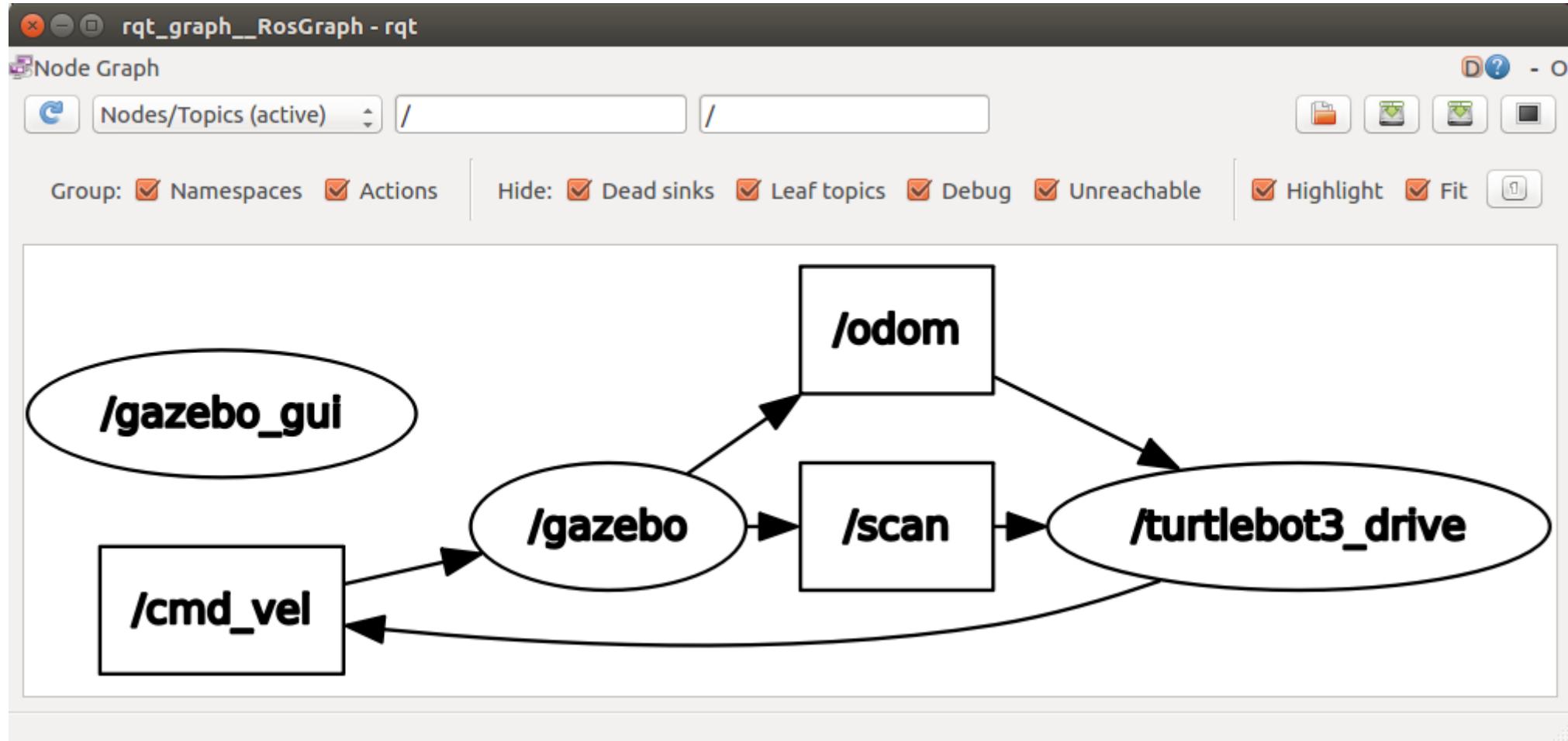


# Turtlebot3 – collision avoidance

---

E' possibile lanciare un nodo per teleoperare il  
nodo mentre il robot si muove in modalità  
di navigazione autonoma?

# Turtlebot3 – rqt\_graph



# Turtlebot3 – RViz

---

RViz può essere usato per visualizzare i topic che vengono pubblicati mentre la simulazione è in esecuzione.

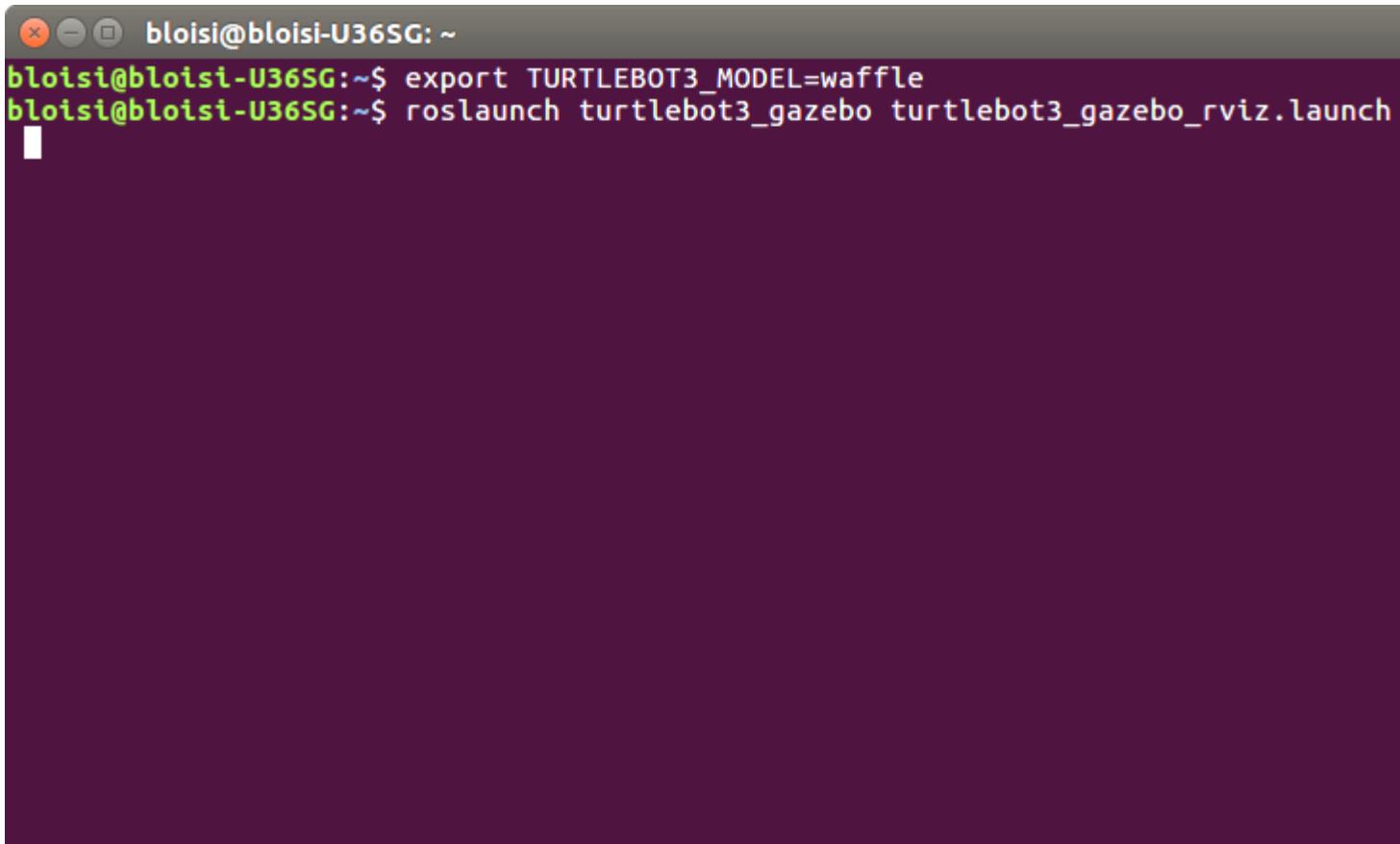
Per lanciare RViz, apriamo un nuovo terminal e digitiamo i comandi seguenti:

```
$ export TURTLEBOT3_MODEL=waffle  
$ roslaunch turtlebot3_gazebo turtlebot3_gazebo_rviz.launch
```

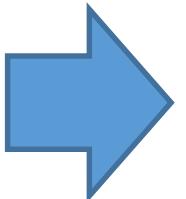
# Turtlebot3 – RViz

---

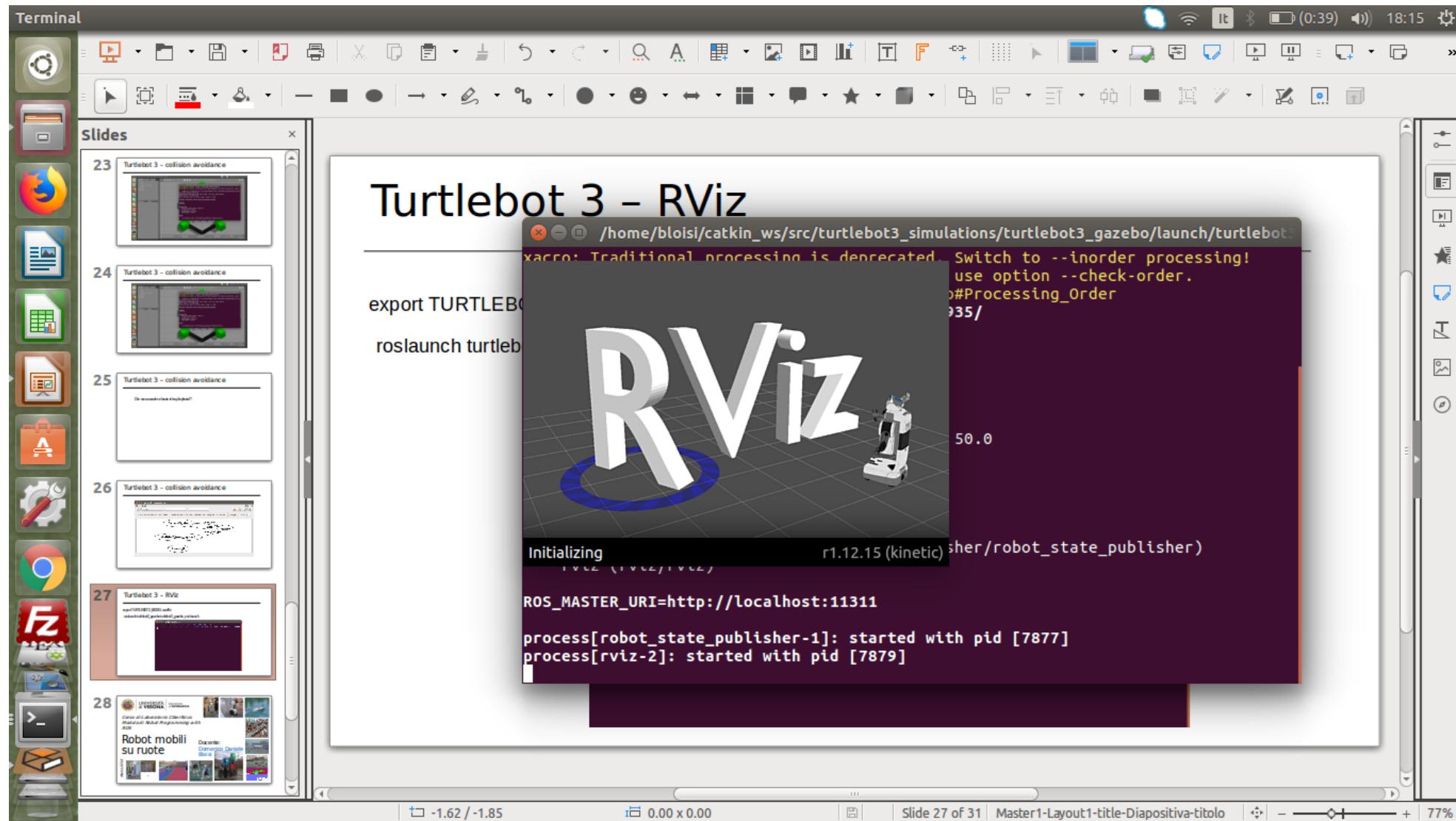
```
$ export TURTLEBOT3_MODEL=waffle  
$ roslaunch turtlebot3_gazebo turtlebot3_gazebo_rviz.launch
```

A screenshot of a terminal window titled "bloisi@bloisi-U36SG:~". The window contains two commands: "export TURTLEBOT3\_MODEL=waffle" and "roslaunch turtlebot3\_gazebo turtlebot3\_gazebo\_rviz.launch".

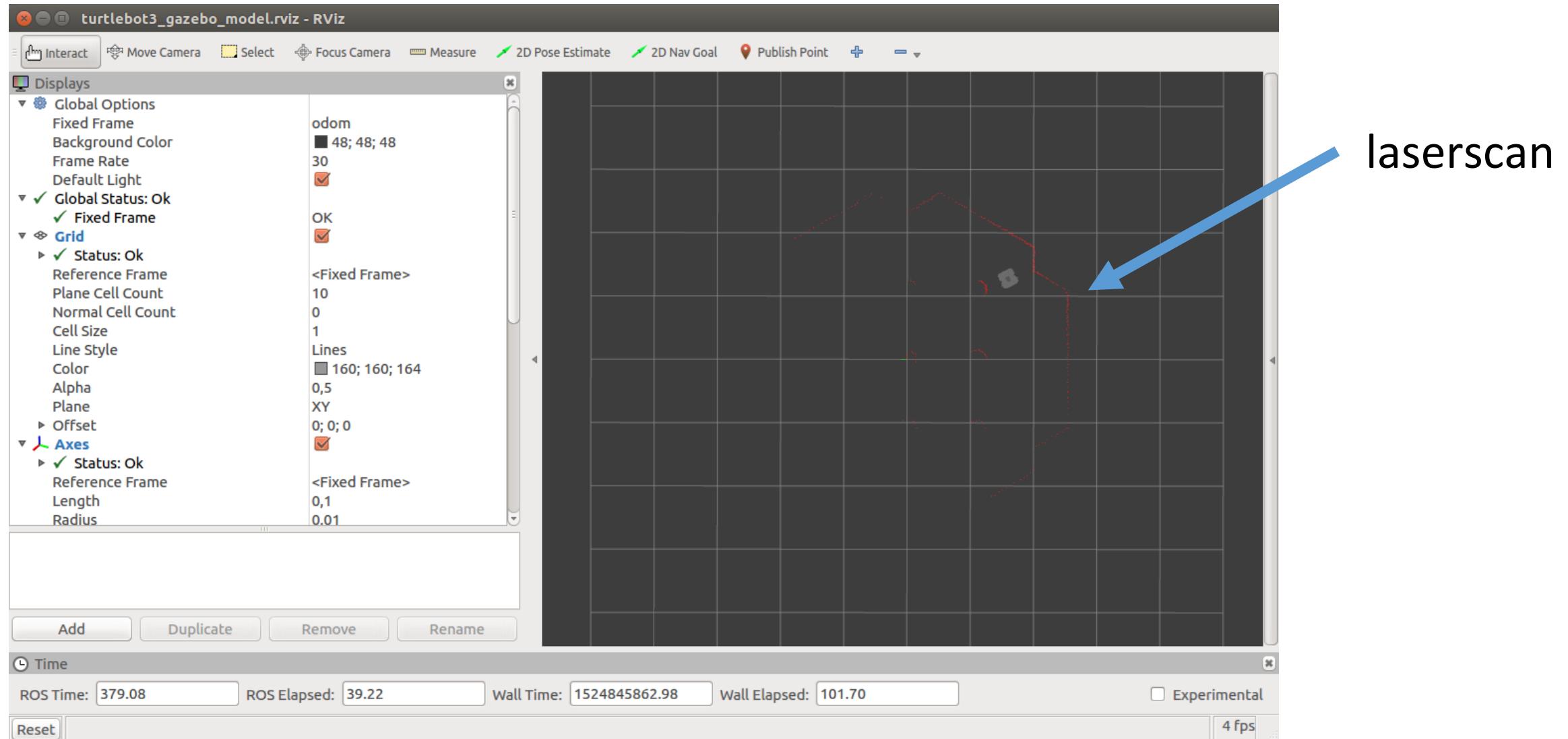
```
bloisi@bloisi-U36SG:~$ export TURTLEBOT3_MODEL=waffle  
bloisi@bloisi-U36SG:~$ roslaunch turtlebot3_gazebo turtlebot3_gazebo_rviz.launch
```



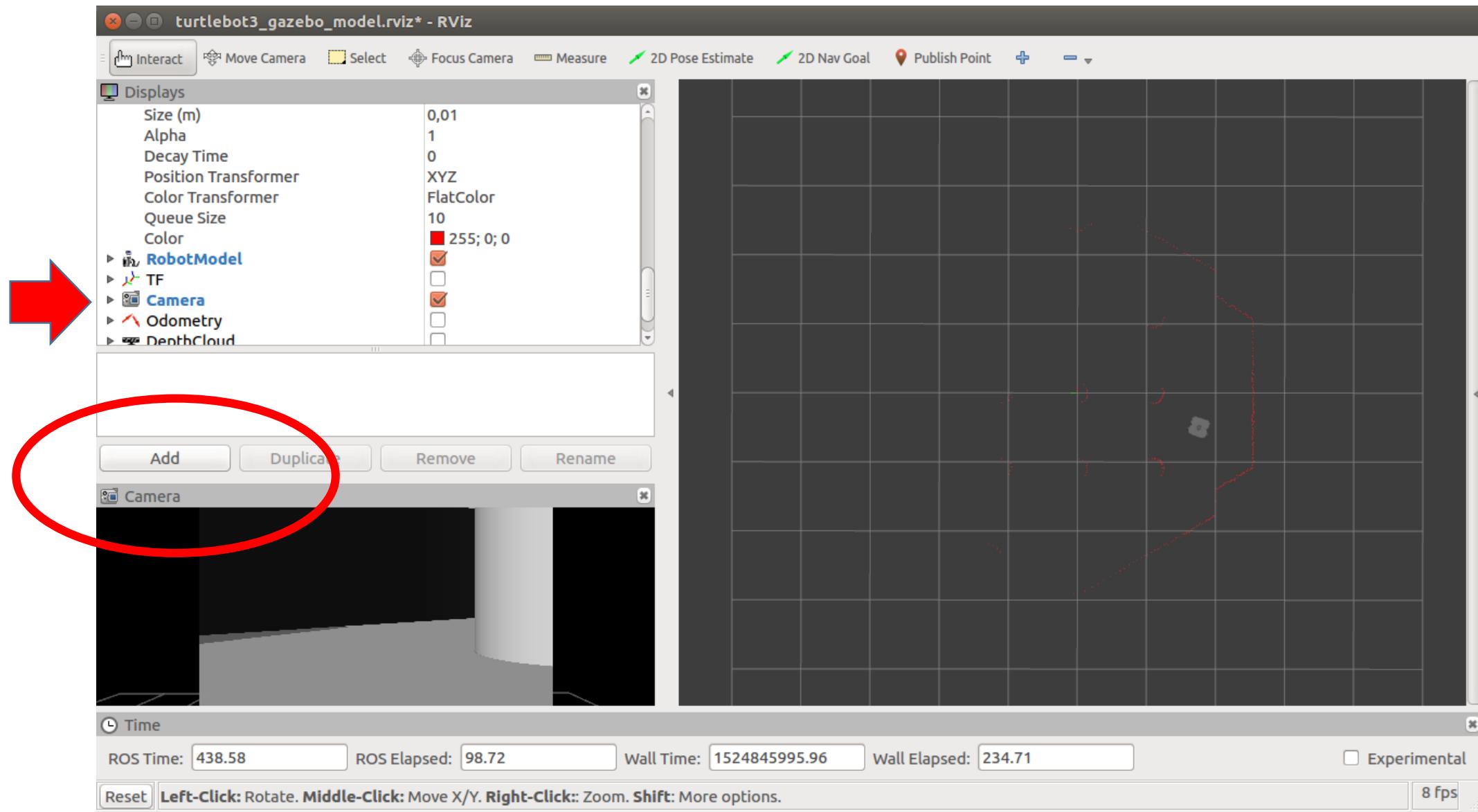
# Turtlebot3 – RViz



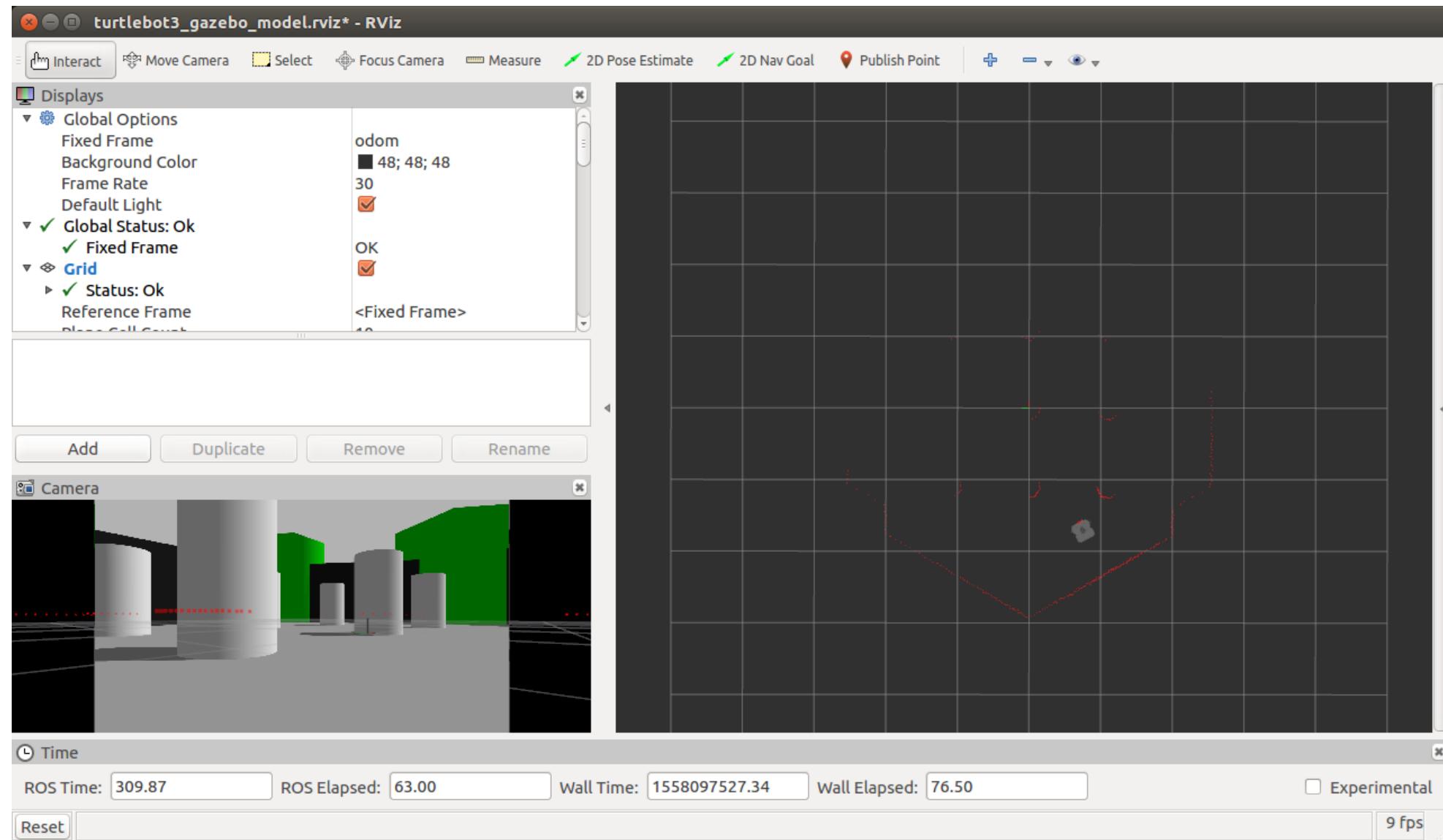
# Turtlebot3 – RViz



# Turtlebot3 – adding camera sensor



# Turtlebot3 – adding camera sensor



# Esercizio – TurtleBot3 House

---

1. Lanciare il nodo per la simulazione della Turtlebot3 House
2. Lanciare la navigazione autonoma del Turtlebot waffle nella Turtlebot3 House
3. Lanciare la teleoperazione da tastiera del robot
4. Visualizzare in RViz i dati provenienti dal laser e dalla telecamera

# Esercizio – cyberlab

The screenshot shows a GitHub repository page for 'dbloisi/cyber\_lab\_gazebo'. The repository has 8 commits, 1 branch, 0 releases, and 1 contributor. The latest commit was on 4 Nov 2017. The repository contains files like README.md, cyber\_lab, README.md, cyber\_lab.world, setup.sh, and turtlebot3\_cyber\_lab.launch.

No description, website, or topics provided.

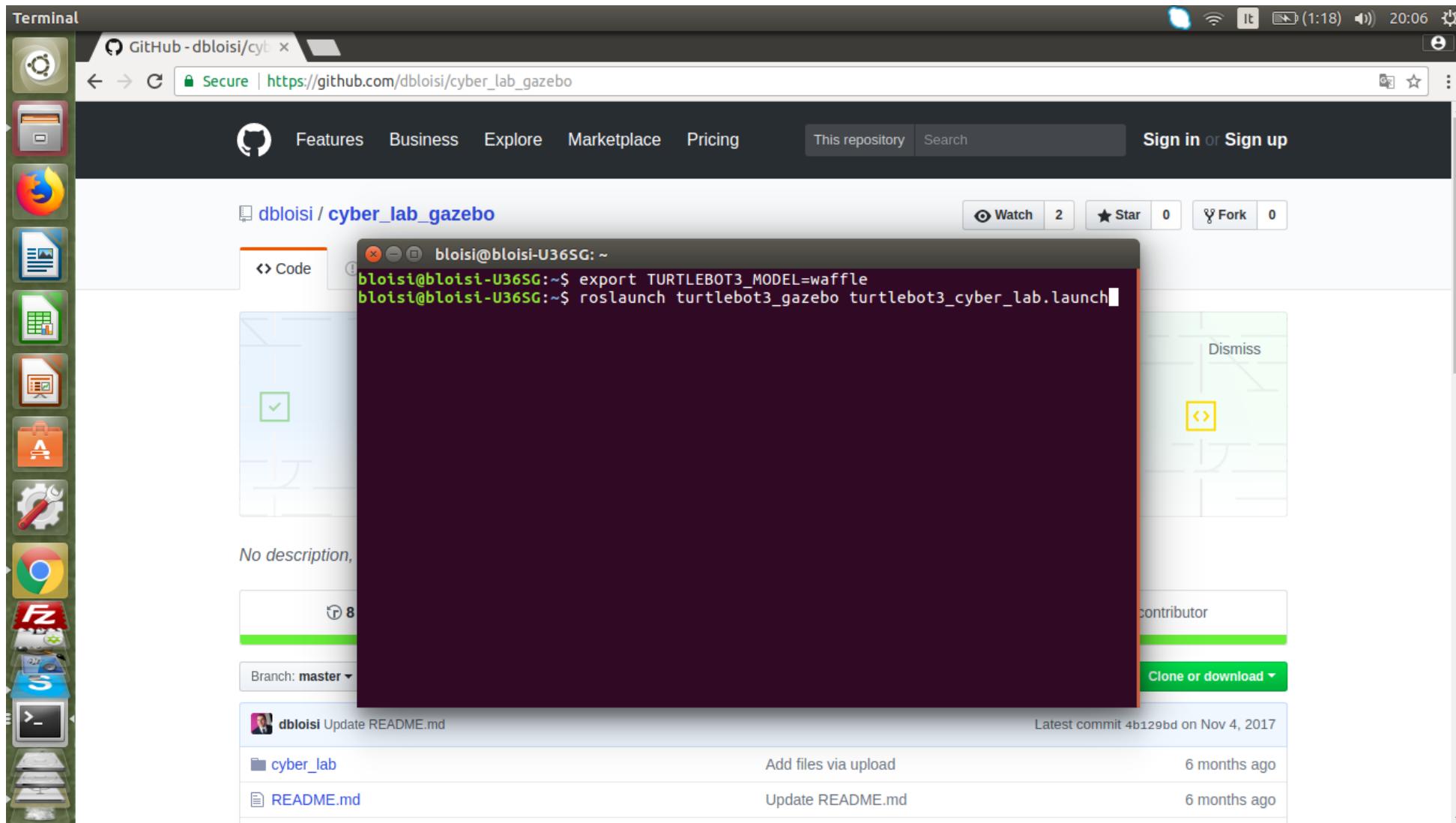
Branch: master ▾ New pull request

Create new file Upload files Find file Clone or download ▾

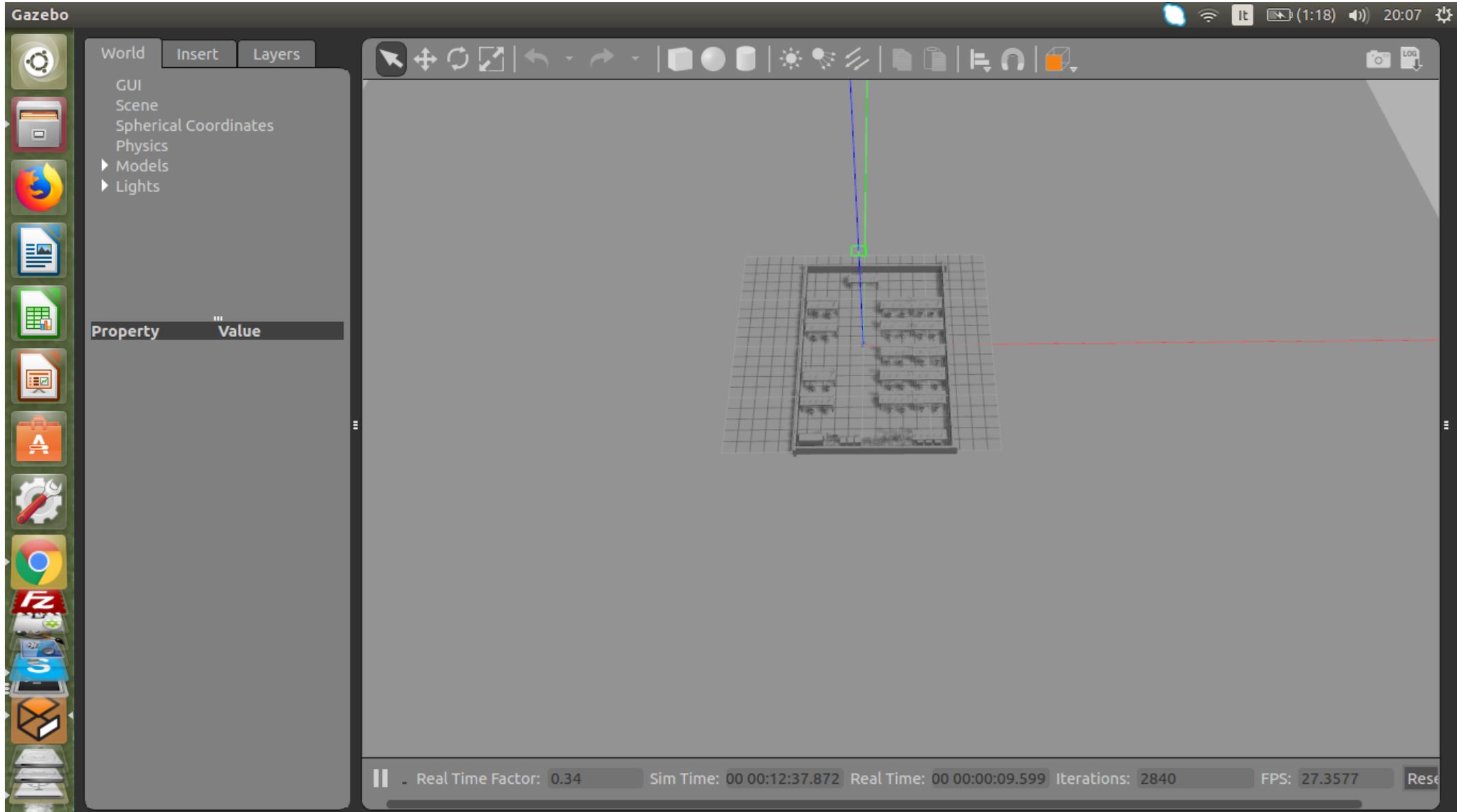
File	Action	Time
dbloisi Update README.md	Latest commit 4b129bd on 4 Nov 2017	
cyber_lab	Add files via upload	6 months ago
README.md	Update README.md	6 months ago
cyber_lab.world	Add files via upload	6 months ago
setup.sh	Add files via upload	6 months ago
turtlebot3_cyber_lab.launch	Add files via upload	6 months ago
README.md		

[https://github.com/dbloisi/cyber\\_lab\\_gazebo](https://github.com/dbloisi/cyber_lab_gazebo)

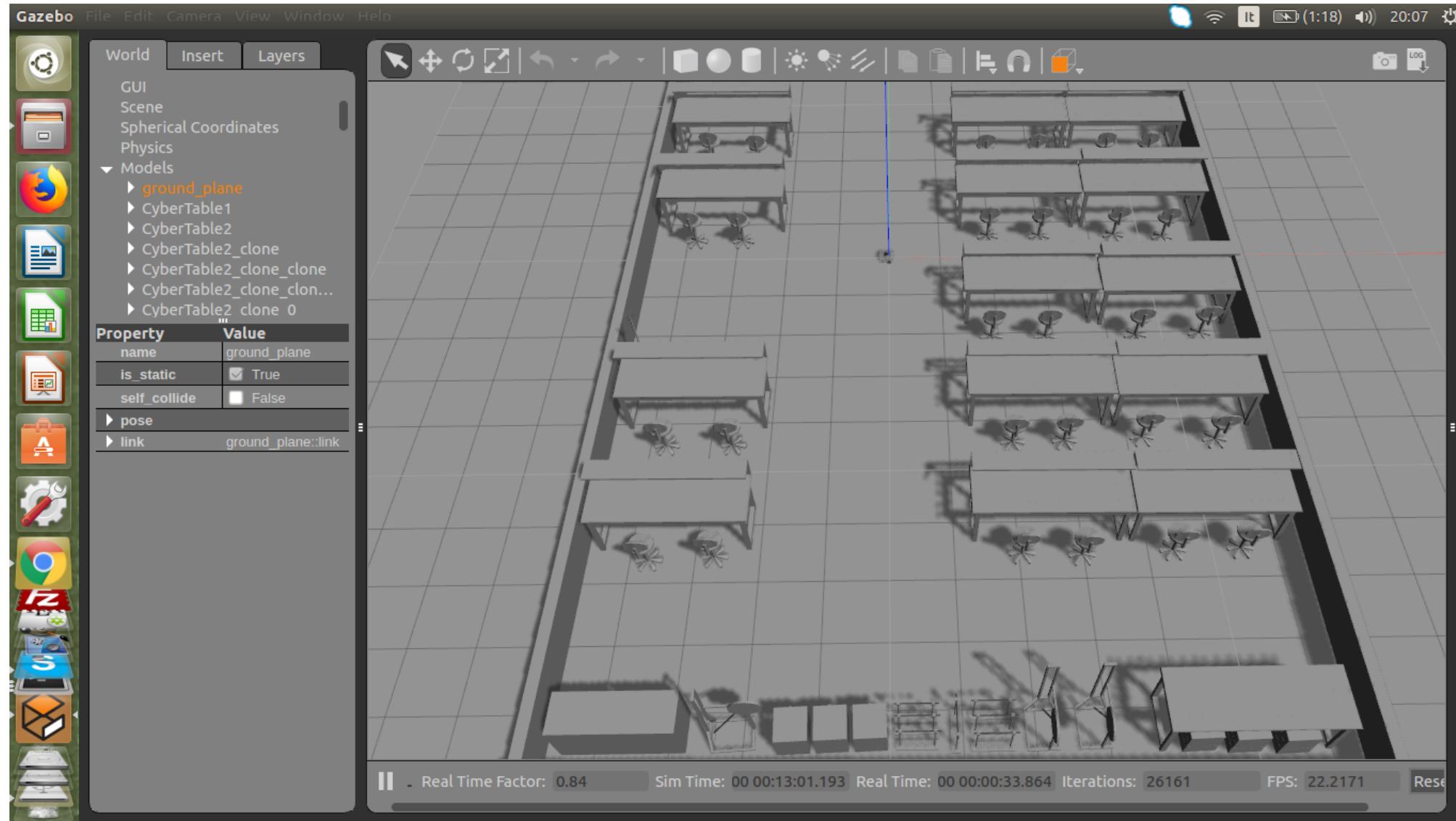
# Esercizio – cyberlab



# Esercizio – cyberlab



# Esercizio – cyberlab





**UNIVERSITÀ DEGLI STUDI  
DELLA BASILICATA**

*Corso di Visione e Percezione*  
A.A. 2019/2020

Docente  
**Domenico Daniele Bloisi**

# Simulatori

Maggio 2020

