PART A: Environment and Software Installations

1. **Robot Operating System (ROS) is an open-source software development kit for robotics applications. ROS offers a standard software platform to developers across industries that will carry them from research and prototyping all the way through to deployment and production.**

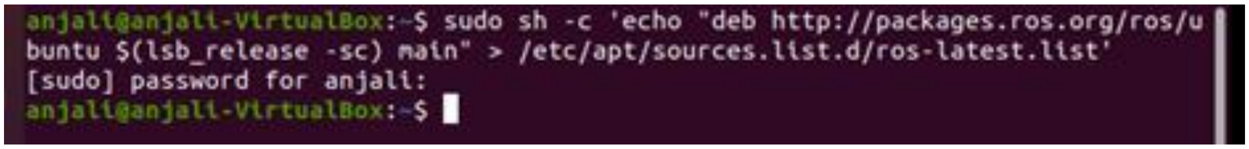
Install ROS 1 in your system as discussed in the LAB. (Help File). Mention the steps in Installation and Configuration.

The steps to install ROS are as follows:

Step 1: First, download the virtual box and install ubuntu.

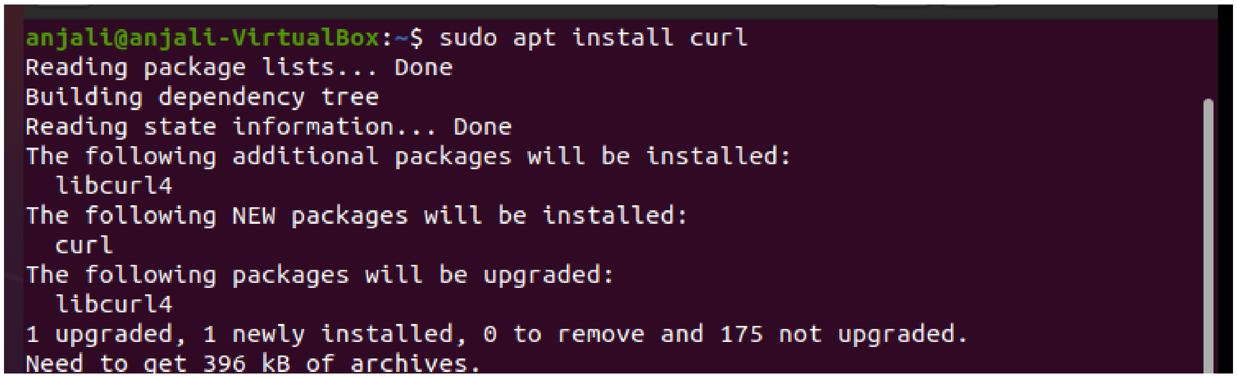
Step 2: Open the terminal type the following commands on the terminal, and run them one after the other.

sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" >/etc/apt/sources.list.d/ros-latest.list'



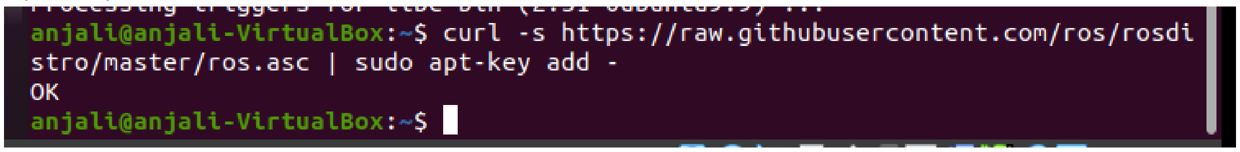
```
anjali@anjali-VirtualBox:~$ sudo sh -c 'echo "deb http://packages.ros.org/ros/u
buntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
[sudo] password for anjali:
anjali@anjali-VirtualBox:~$
```

sudo apt install curl



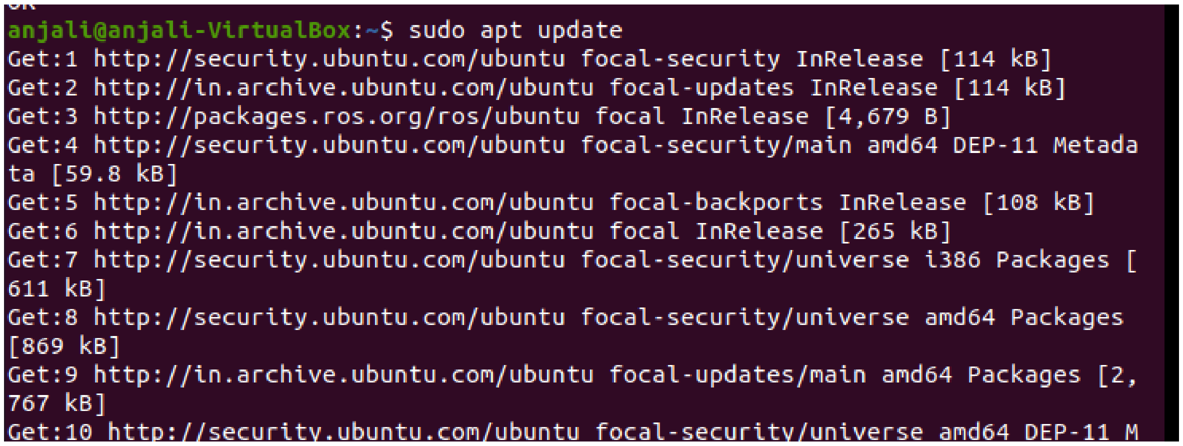
```
anjali@anjali-VirtualBox:~$ sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl
The following packages will be upgraded:
  libcurl4
1 upgraded, 1 newly installed, 0 to remove and 175 not upgraded.
Need to get 396 kB of archives.
```

curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add –



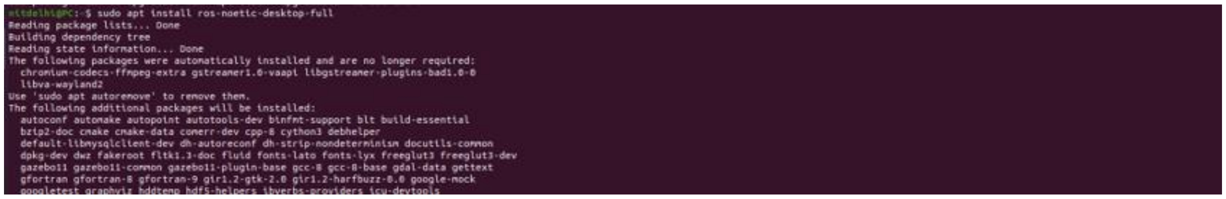
```
anjali@anjali-VirtualBox:~$ curl -s https://raw.githubusercontent.com/ros/rosdi
stro/master/ros.asc | sudo apt-key add -
OK
anjali@anjali-VirtualBox:~$
```

sudo apt update



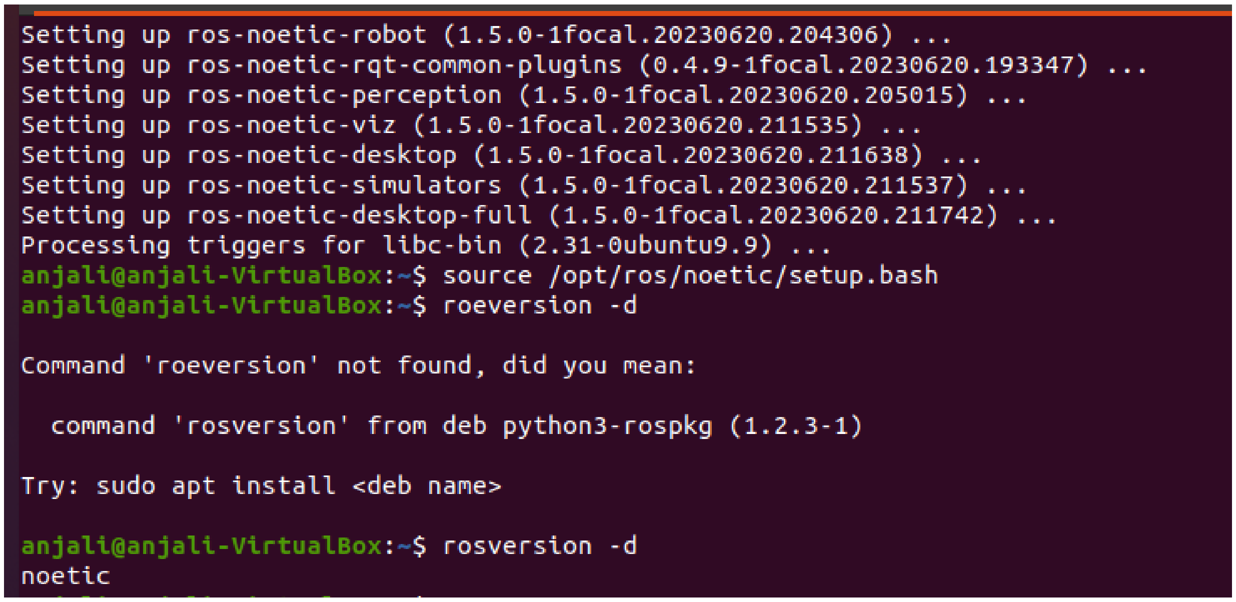
```
anjali@anjali-VirtualBox:~$ sudo apt update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://packages.ros.org/ros/ubuntu focal InRelease [4,679 B]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metada
ta [59.8 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/universe i386 Packages [
611 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages
[869 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2,
767 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 M
```

sudo apt install ros-noetic-desktop-full



```
nitdelhi@PC:~$ sudo apt install ros-noetic-desktop-full
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0
  libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  autoconf automake autopoint autotools-dev binfmt-support blt build-essential
  bzip2-doc cmake cmake-data comerr-dev cpp-8 cython3 debhelper
  default-libmysqlclient-dev dh-autoreconf dh-strip-nondeterminism docutils-common
  dpkg-dev dwz fakeroot fltk1.3-doc fluid fonts-lato fonts-lyx freeglut3 freeglut3-dev
  gazebo11 gazebo11-common gazebo11-plugin-base gcc-8 gcc-8-base gdal-data gettext
  gfortran gfortran-8 gfortran-9 gir1.2-gtk-2.0 gir1.2-harfbuzz-0.0 google-mock
  googletest graphviz hddtemp hdf5-helpers ibverbs-providers icu-devtools
```

source /opt/ros/noetic/setup.bash



```
Setting up ros-noetic-robot (1.5.0-1focal.20230620.204306) ...
Setting up ros-noetic-rqt-common-plugins (0.4.9-1focal.20230620.193347) ...
Setting up ros-noetic-perception (1.5.0-1focal.20230620.205015) ...
Setting up ros-noetic-viz (1.5.0-1focal.20230620.211535) ...
Setting up ros-noetic-desktop (1.5.0-1focal.20230620.211638) ...
Setting up ros-noetic-simulators (1.5.0-1focal.20230620.211537) ...
Setting up ros-noetic-desktop-full (1.5.0-1focal.20230620.211742) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
anjali@anjali-VirtualBox:~$ source /opt/ros/noetic/setup.bash
anjali@anjali-VirtualBox:~$ roeversion -d

Command 'roeversion' not found, did you mean:

  command 'rosversion' from deb python3-rospkg (1.2.3-1)

Try: sudo apt install <deb name>

anjali@anjali-VirtualBox:~$ rosversion -d
noetic
```

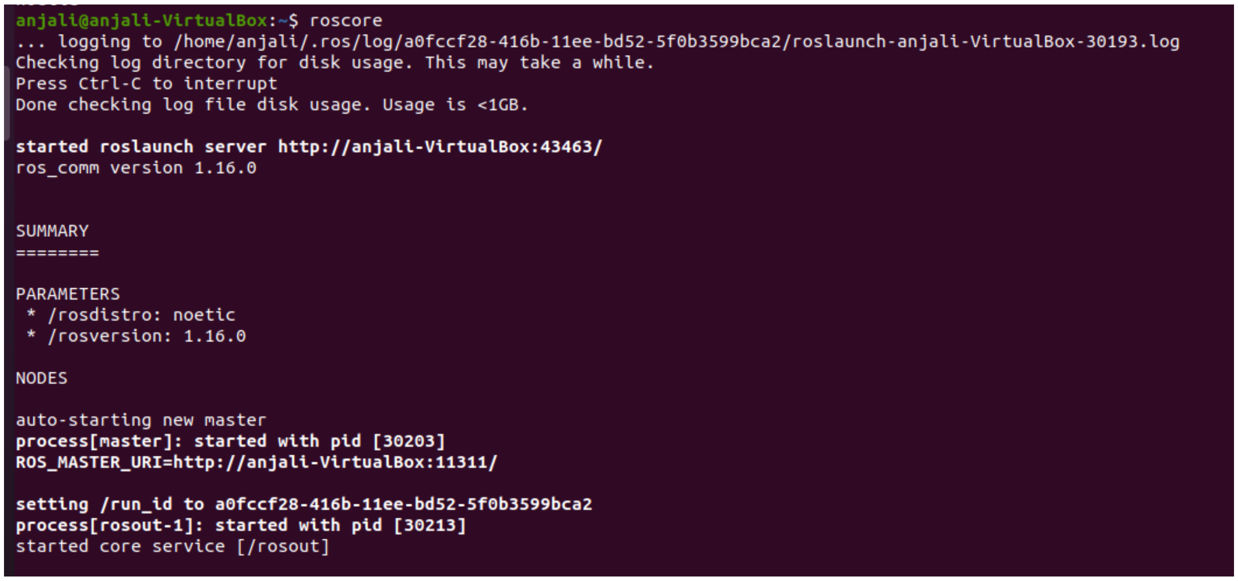
Step 3: ROS is installed and configured now type roscore command on the terminal and start executing the command on a new terminal.

Observation: Ubuntu has been the primary platform for ROS from the very beginning, thanks to its flexibility and user-friendliness. ROS is led by Open Robotics, similar to how Canonical supports Ubuntu

2. **Execute the turtlesim node as discussed in LAB to control the Turtlesim by terminal and python script; and implement the basics of 2D navigation.**

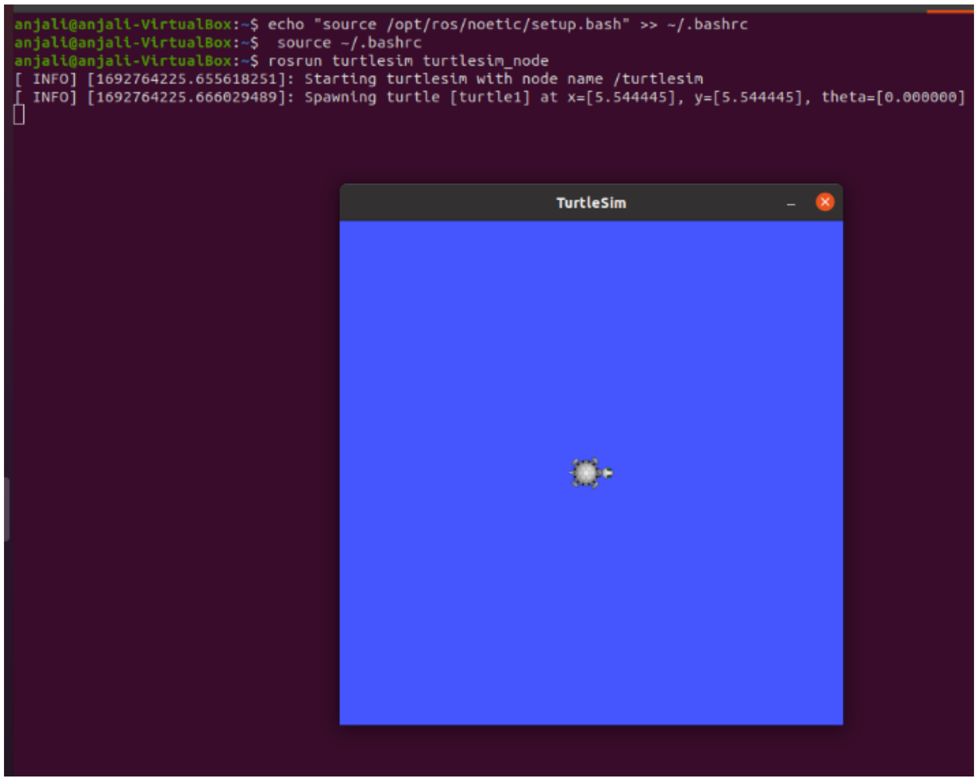
- **Explore the ROS Basic Commands : roscore, rosrun, rosnode and rostopic.**

Roscore command is used to start the ros master which is necessary to start the ros.



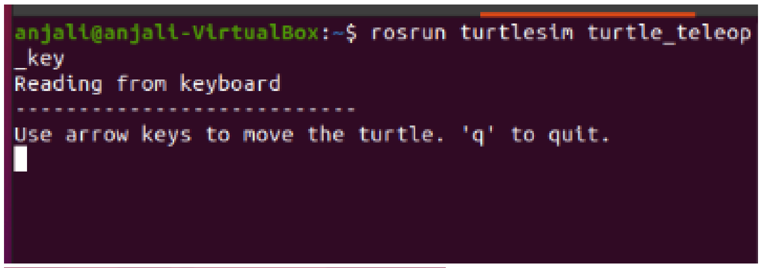
```
anjali@anjali-VirtualBox:~$ roscore
... logging to /home/anjali/.ros/log/a0fccf28-416b-11ee-bd52-5f0b3599bca2/roslaunch-anjali-VirtualBox-30193.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://anjali-VirtualBox:43463/
ros_comm version 1.16.0


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.16.0

NODES

auto-starting new master
process[master]: started with pid [30203]
ROS_MASTER_URI=http://anjali-VirtualBox:11311/

setting /run_id to a0fccf28-416b-11ee-bd52-5f0b3599bca2
process[rosout-1]: started with pid [30213]
started core service [/rosout]
```

Rosrun command is used to directly run a node in a package.

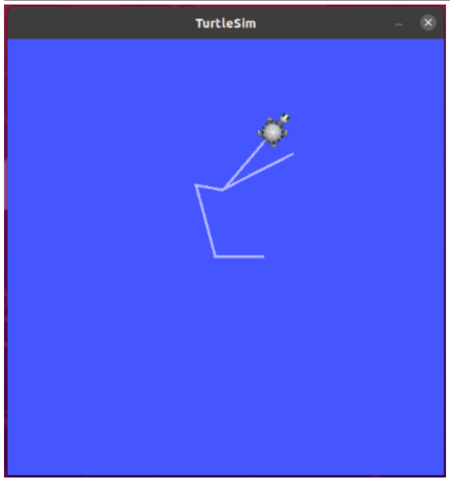


```
anjali@anjali-VirtualBox:~$ echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
anjali@anjali-VirtualBox:~$  source ~/.bashrc
anjali@anjali-VirtualBox:~$ rosrun turtlesim turtlesim_node
[ INFO] [1692764225.655618251]: Starting turtlesim with node name /turtlesim
[ INFO] [1692764225.666029489]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]
```

To move the turtle one can use the following command.

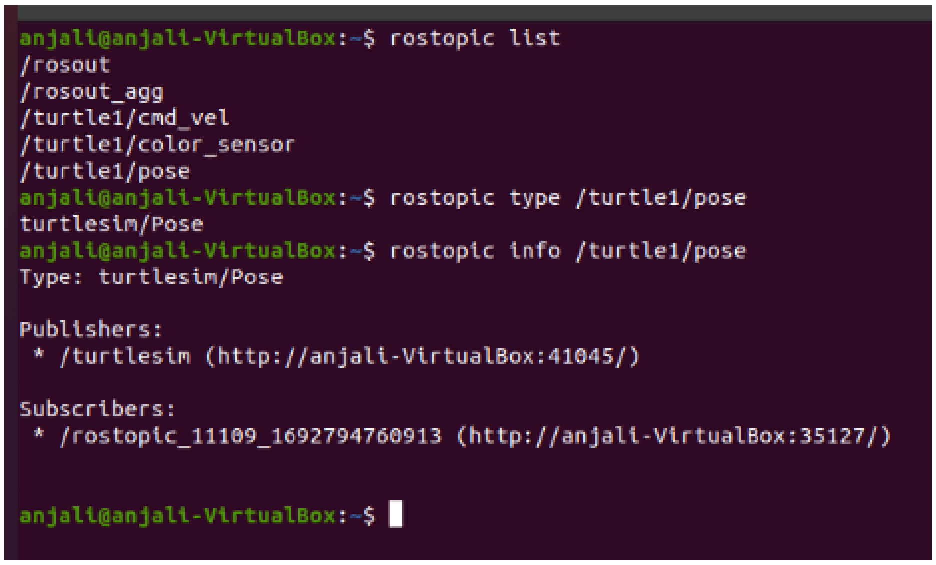


```
anjali@anjali-VirtualBox:~$ rosrun turtlesim turtle_teleop
_key
Reading from keyboard
---------------------------
Use arrow keys to move the turtle. 'q' to quit.
```



Now you can use arrow keys to move the turtle.

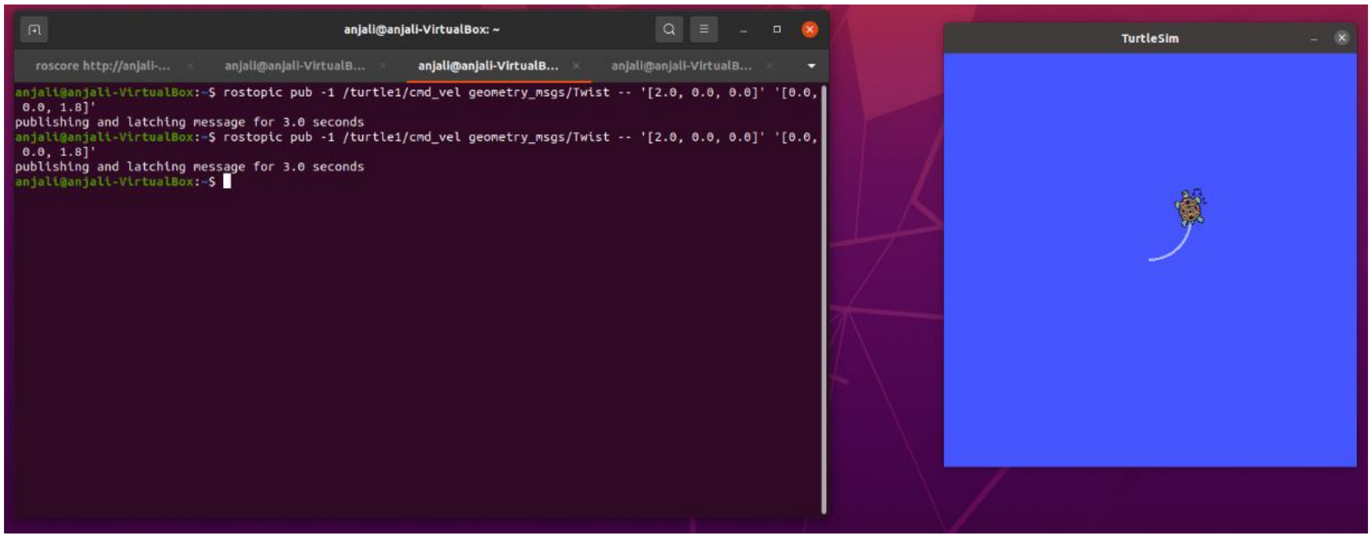
The rostopic command displays debug information about ROS Topics, including publishers, subscribers, publishing rate, and ROS Messages.



```
anjali@anjali-VirtualBox:~$ rostopic list
/rosout
/rosout_agg
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
anjali@anjali-VirtualBox:~$ rostopic type /turtle1/pose
turtlesim/Pose
anjali@anjali-VirtualBox:~$ rostopic info /turtle1/pose
Type: turtlesim/Pose

Publishers:
 * /turtlesim (http://anjali-VirtualBox:41045/)

Subscribers:
 * /rostopic_11109_1692794760913 (http://anjali-VirtualBox:35127/)


anjali@anjali-VirtualBox:~$
```

- **Send a single message to turtlesim telling it to move with a linear velocity of 2.0, and an angular velocity of 1.8. It will move from its starting position along a circular trajectory for a distance and then stop.**

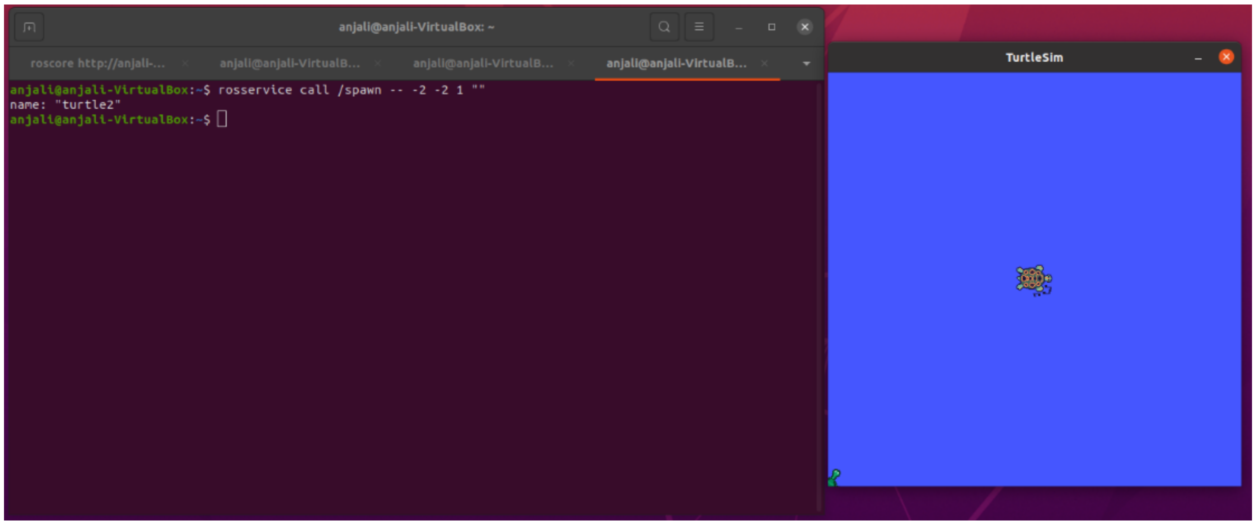
Type the following command on terminal after running turtlesim:
Rostopic pub -1 /turtlesim/cmd_vel geometry_msgs/Twist .. ‘[2.0, 0.0, 0.0]’ ‘[0.0, 0.0, 1.8]’


anjali@anjali-VirtualBox: ~
roscore http://anjali-...
anjali@anjali-VirtualB...
anjali@anjali-VirtualB...
anjali@anjali-VirtualB...
TurtleSim

```
anjali@anjali-VirtualBox:~$ rostopic pub -1 /turtle1/cmd_vel geometry_msgs/Twist -- '[2.0, 0.0, 0.0]' '[0.0,
 0.0, 1.8]'
publishing and latching message for 3.0 seconds
anjali@anjali-VirtualBox:~$ rostopic pub -1 /turtle1/cmd_vel geometry_msgs/Twist -- '[2.0, 0.0, 0.0]' '[0.0,
 0.0, 1.8]'
publishing and latching message for 3.0 seconds
anjali@anjali-VirtualBox:~$
```

- **Try to add a new turtle. You don’t have to stop the rosnode turtlesim. Move the second turtle to position -2 -2 1 (x y theta) by the terminal.**

Run the following command on command line:
Rosservice call /spawn -- -2 -2 1 “”


anjali@anjali-VirtualBox: ~
roscore http://anjali-...
anjali@anjali-VirtualB...
anjali@anjali-VirtualB...
anjali@anjali-VirtualB...
TurtleSim

```
anjali@anjali-VirtualBox:~$ rosservice call /spawn -- -2 -2 1 ""
name: "turtle2"
anjali@anjali-VirtualBox:~$
```

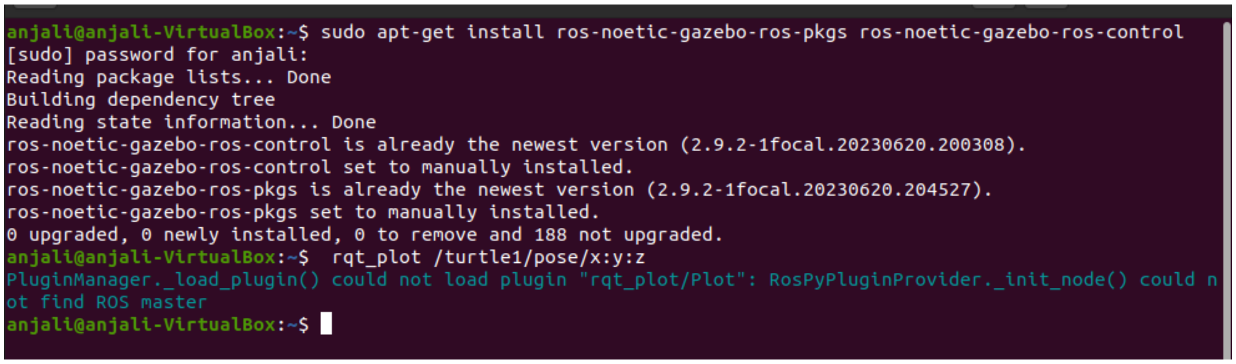
Observation: ROS commands are easy to run and understand. Turtlesim is a tool made for teaching ROS and ROS packages. It is a lightweight simulator for learning ROS 2. It illustrates what ROS 2 does at the most basic level to give you an idea of what you will do with a real robot or a robot simulation.

3. **Install Gazebo in your system and mention the steps in installation and configuration.**

The steps to install Gazebo are as follows:

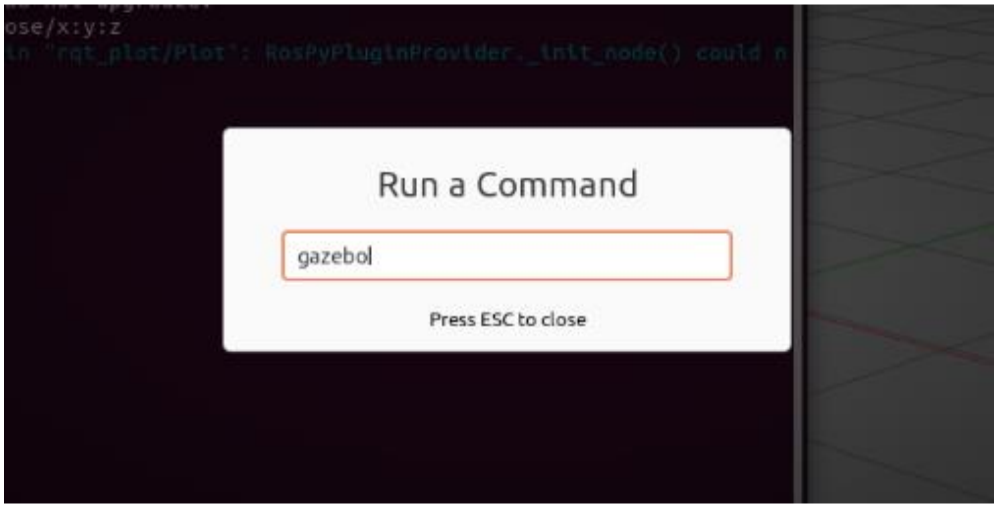
Step 1: Type the following command on the terminal to install gazebo.

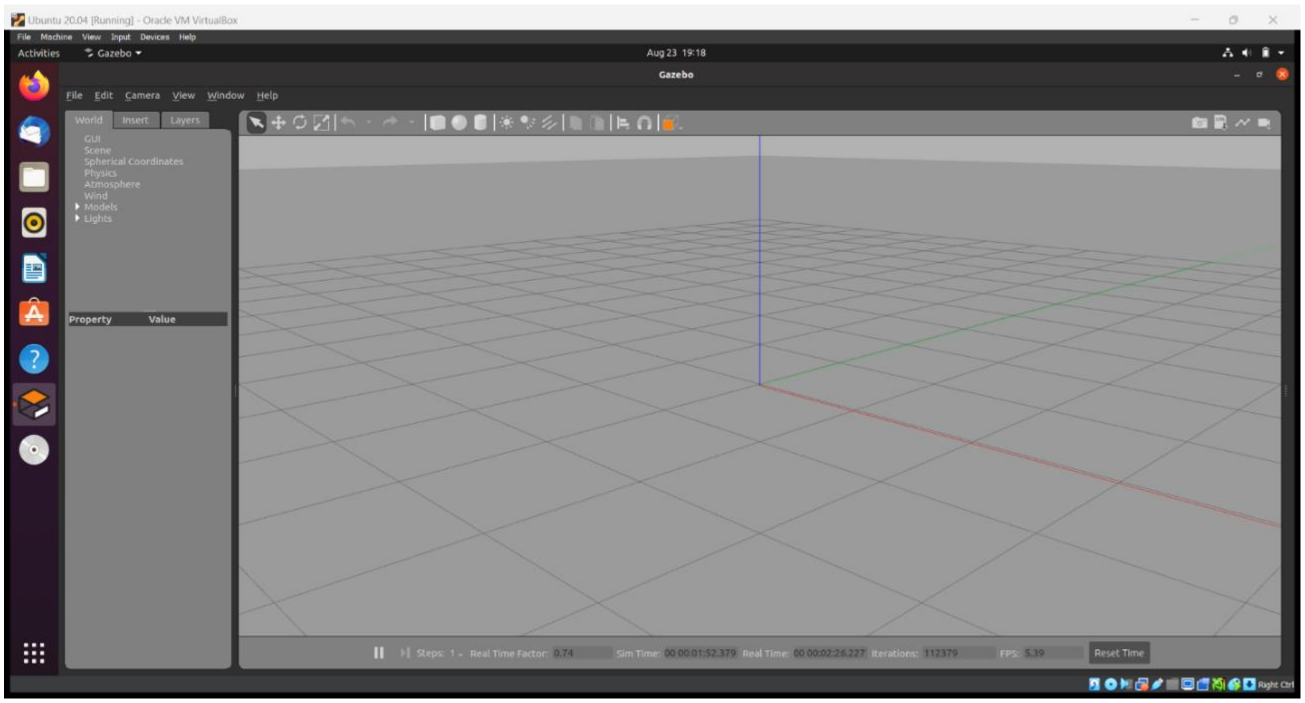
sudo apt-get install ros-noetic-gazebo-ros-pkgs ros-noetic-gazebo-ros-control



```
anjali@anjali-VirtualBox:~$ sudo apt-get install ros-noetic-gazebo-ros-pkgs ros-noetic-gazebo-ros-control
[sudo] password for anjali:
Reading package lists... Done
Building dependency tree
Reading state information... Done
ros-noetic-gazebo-ros-control is already the newest version (2.9.2-1focal.20230620.200308).
ros-noetic-gazebo-ros-control set to manually installed.
ros-noetic-gazebo-ros-pkgs is already the newest version (2.9.2-1focal.20230620.204527).
ros-noetic-gazebo-ros-pkgs set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 188 not upgraded.
anjali@anjali-VirtualBox:~$  rqt_plot /turtle1/pose/x:y:z
PluginManager._load_plugin() could not load plugin "rqt_plot/Plot": RosPyPluginProvider._init_node() could n
ot find ROS master
anjali@anjali-VirtualBox:~$ 
```

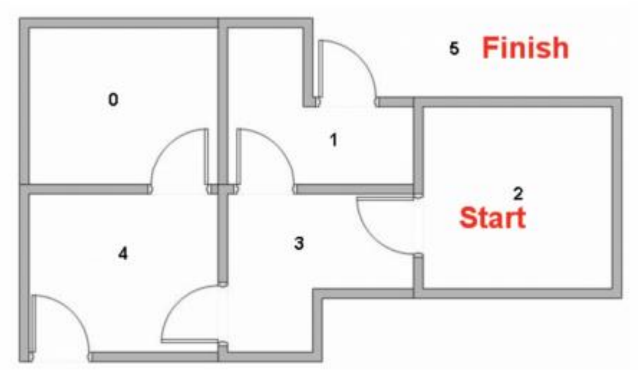
Step 2: Now you can run gazebo as it is installed and configured successfully.



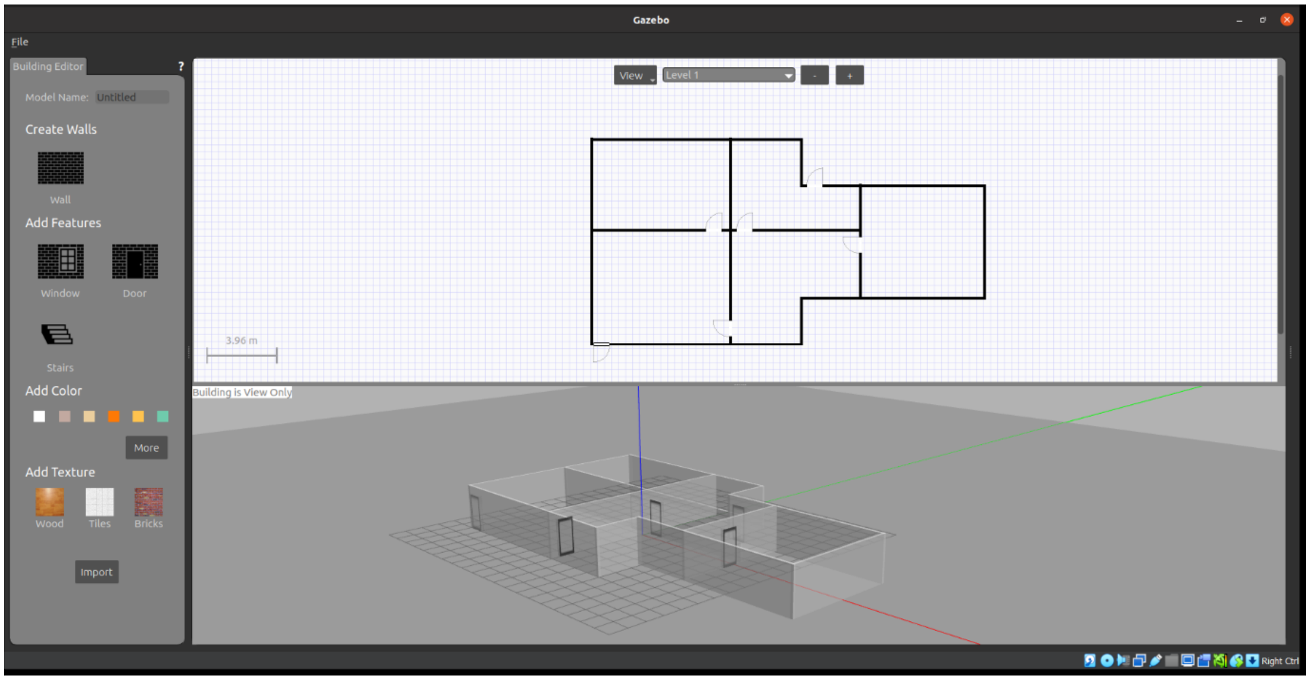


Observation: Gazebo robot simulation tool is easy to install and use. Gazebo brings a fresh approach to simulation with a complete toolbox of development libraries and cloud services to make simulation easy.

4. Construct the following house configuration space/ environment using Gazebo.



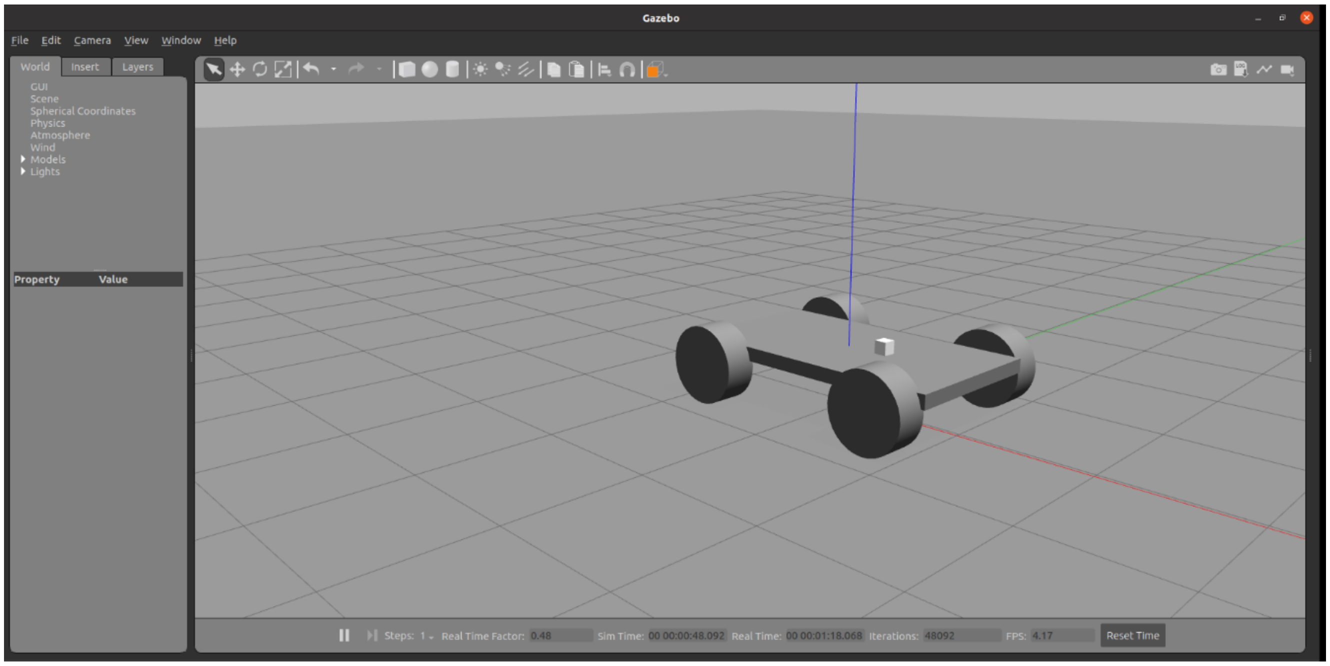
- Go to the building editor view, build the model as shown in the figure and then save the model as a room_setup.model file.
- Exit the model.
- Then save the world as room_setup.world file for using the environment further.



Observation: Building or environment in gazebo is easy. An environment can be build by using the building editor view.

5. **Construct a wheeled differential robot in the environment of Gazebo. Move the robot in one direction. Show your creativity in the designing of the robot.**

To construct a wheeled differential model first go to the model editor.
There we can use various shapes to construct the model in the shape and size we want.
After constructing various parts of the model. Join the model using the joint command.
After that we can exit the model editor mode and then simulate to see the model moving in one direction.



Steps: 1 Real Time Factor: 0.48 Sim Time: 00 00:00:48.092 Real Time: 00 00:01:18.068 Iterations: 48092 FPS: 4.17

Observation: We can build a model easily in gazebo using model editor option in the view option given in the taskbar.

PART B : Exploratory Problem

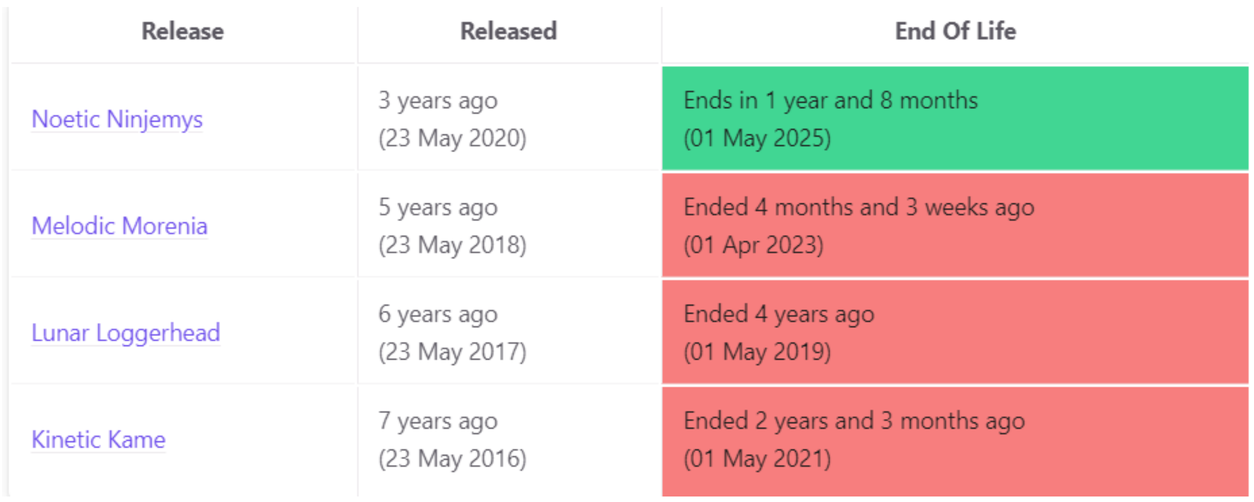
6. **Explore the**

- **History of ROS.**

While attending Stanford, Keenan Wyrobek and Eric Berger began working on ROS as a side project to solve the robotics industry's problem of constantly reinventing the wheel. Those two men were concerned about the most typical robotics issue at the time:
Re-implementing the software infrastructure necessary to develop complex robotics algorithms which takes an excessive amount of time
Too little effort was spent developing those infrastructure-based intelligent robots programs.
The Stanford Personal Robotics Programme was established in 2006 by Eric and Keenan to address this issue. Its goal was to provide a framework that enabled processes to communicate with one another as well as some tools to aid in the creation of code on top of that framework. All of that infrastructure was intended to be used to generate code for the Personal Robot, a robot they would also build and utilise as a testbed and an example for others. To enable universities to create software using their architecture, they would construct ten of such robots and provide them to the institutions.
People who are more familiar with ROS will recognise the Rviz, rqt_tools, and other early versions of present ROS releases as well as the ROS-comm libraries. Also, the Personal Robot was the precursor of the famous PR2 robot.

- **ROS Versions.**

ROS currently releases a version every year in May, following the release of Ubuntu LTS versions. Currently, two active major versions are seeing releases: ROS 1 and ROS 2. Aside to this there is the ROS-Industrial or ROS-I derivate project since at least 2012.
In ROS1, you've been used to write launch files with XML. In ROS2 you will now use Python to write your launch files.
Some of the ROS releases are as follows:



Release	Released	End Of Life
Noetic Ninjemys	3 years ago (23 May 2020)	Ends in 1 year and 8 months (01 May 2025)
Melodic Morenia	5 years ago (23 May 2018)	Ended 4 months and 3 weeks ago (01 Apr 2023)
Lunar Loggerhead	6 years ago (23 May 2017)	Ended 4 years ago (01 May 2019)
Kinetic Kame	7 years ago (23 May 2016)	Ended 2 years and 3 months ago (01 May 2021)

- **Examples of ROS-compatible robots and hardware.**

ROS compatible robots:

- Turtlebot3: robot used in education, research, hobby, and product prototyping.
- Husky: robot developed (and integrated into ROS) by Clearpath Robotics
- Shadow Robot Hand: a fully dexterous humanoid hand.
- PR1: personal robot developed in Ken Salisbury's lab at Stanford.
- GoPiGo3: Raspberry Pi-based educational robot, supports ROS.

ROS-compatible hardware:

- Raspberry Pi: image of Ubuntu Mate with ROS by Ubiquity Robotics.
- BeagleBoard: the robotics lab of the Katholieke Universiteit Leuven, Belgium has ported ROS to the Beagleboard.

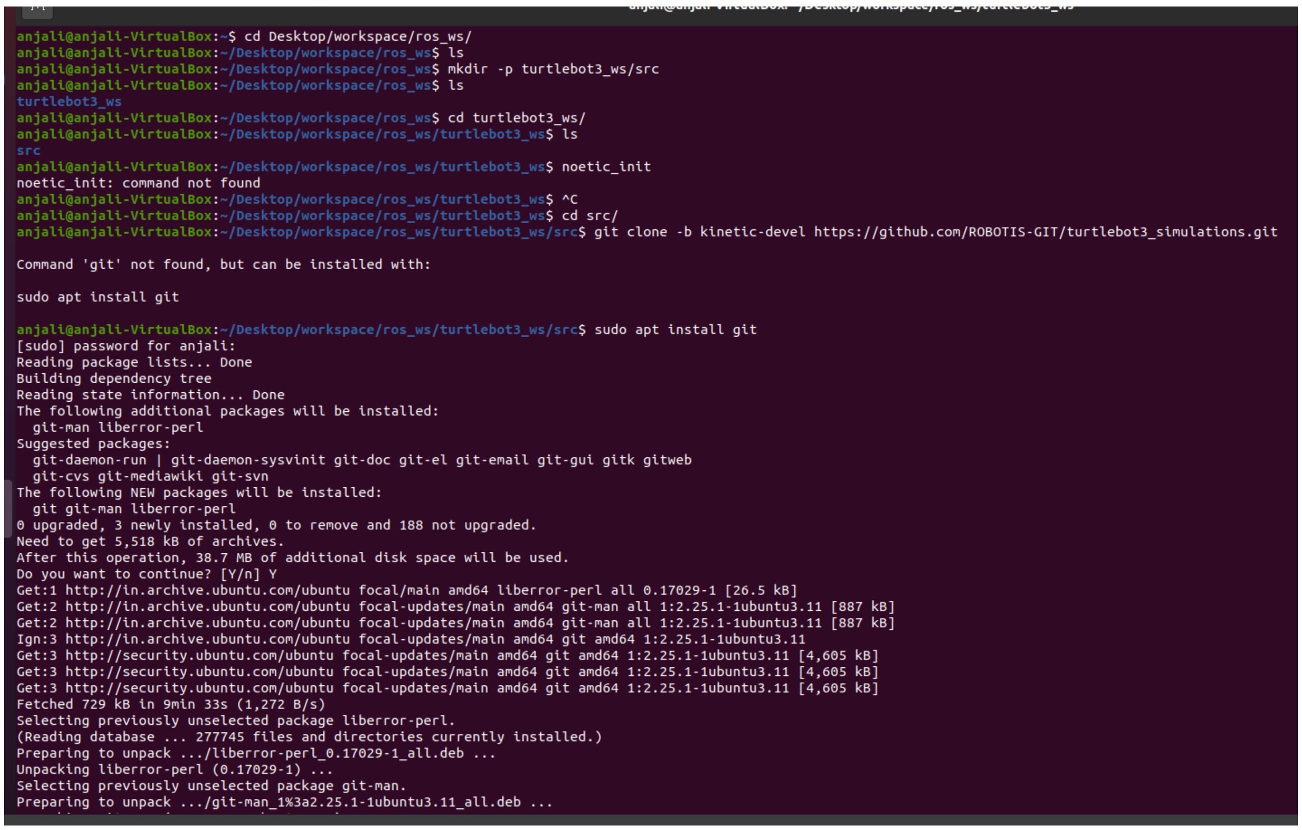
7. **Explore the different robot simulator used for research, design, and development of robots.**

Some of the robot simulators used for design, research, and development are as follows:

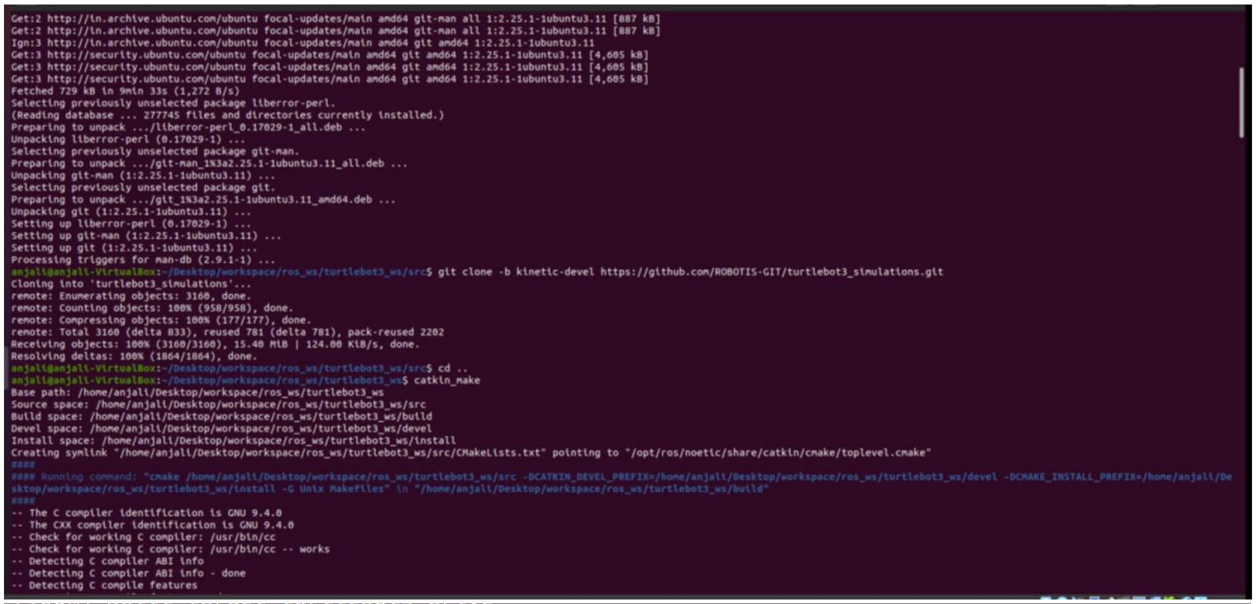
- USARSim: It is an open source robot simulator that can be used both for research and education. Most notably, it constitutes the simulation engine used to run the virtual robots competition within the Robocup initiative.
- RoboDK: It is an offline programming and simulation software for industrial robots.
- Webots: It is a free and open-source 3D robot simulator used in industry, education and research. Webots offers the possibility to take screenshots and record simulations.
- SimSpark: It is a generic simulation system for various multiagent simulations. It supports developing physical AI and robotics research simulations with an open-source application framework. It is commonly used in academic research and education.
- CoppeliaSim: It is a robot simulator used in industry, education and research. It uses a kinematics engine for forward and inverse kinematics calculations, and several physics simulation libraries to perform rigid body simulation.

8. **Explore the Turtle bot3 / Husky in the Gazebo in the 5 room setup (created in problem no 4) and discussed in the lab.**
   - Firstly create a new folder on desktop and name it workspace.
   - Then create a folder inside the workspace and name it ros_ws
   - Inside ros_ws create a folder named as turtlebot3_ws.
   - Inside turtlebot3_ws create a folder src
   - Now run the following commands in terminal:

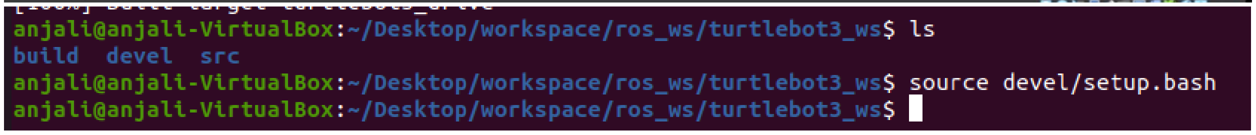
```
cd Desktop/workspace/ros_ws/turtlebot3_ws/src
git clone -b kinetic-devel https://github.com/ROBOTISGIT/turtlebot3_simulations.git
cd ..
catkin_make
source devel/setup.bash
export TURTLEBOT3_MODEL=burger
```



```
anjali@anjali-VirtualBox:~$ cd Desktop/workspace/ros_ws/
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws$ ls
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws$ mkdir -p turtlebot3_ws/src
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws$ ls
turtlebot3_ws
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws$ cd turtlebot3_ws/
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws$ ls
src
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws$ noetic_init
noetic_init: command not found
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws$ ^C
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws$ cd src/
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws/src$ git clone -b kinetic-devel https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git

Command 'git' not found, but can be installed with:

sudo apt install git

anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws/src$ sudo apt install git
[sudo] password for anjali:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb
  git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 188 not upgraded.
Need to get 5,518 kB of archives.
After this operation, 38.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all 1:2.25.1-1ubuntu3.11 [887 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all 1:2.25.1-1ubuntu3.11 [887 kB]
Ign:3 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:2.25.1-1ubuntu3.11
Get:3 http://security.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:2.25.1-1ubuntu3.11 [4,605 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:2.25.1-1ubuntu3.11 [4,605 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:2.25.1-1ubuntu3.11 [4,605 kB]
Fetched 729 kB in 9min 33s (1,272 B/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 277745 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.25.1-1ubuntu3.11_all.deb ...
```



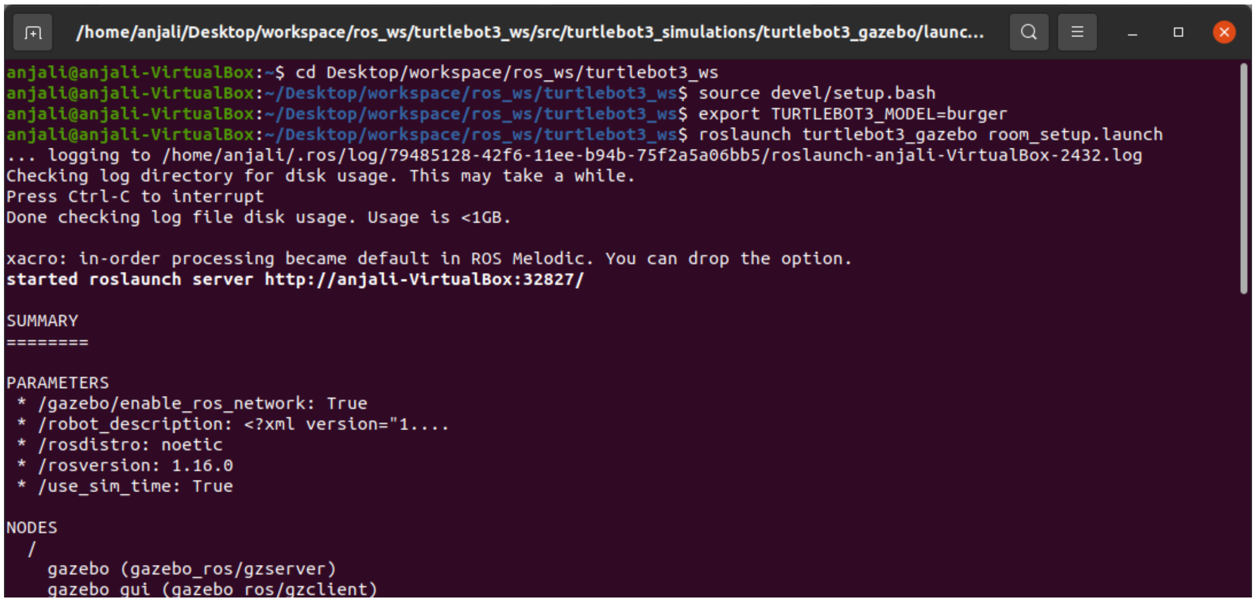
```
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all 1:2.25.1-1ubuntu3.11 [887 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all 1:2.25.1-1ubuntu3.11 [887 kB]
Ign:3 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:2.25.1-1ubuntu3.11
Get:3 http://security.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:2.25.1-1ubuntu3.11 [4,605 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:2.25.1-1ubuntu3.11 [4,605 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:2.25.1-1ubuntu3.11 [4,605 kB]
Fetched 729 kB in 9min 33s (1,272 B/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 277745 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.25.1-1ubuntu3.11_all.deb ...
Unpacking git-man (1:2.25.1-1ubuntu3.11) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.25.1-1ubuntu3.11_amd64.deb ...
Unpacking git (1:2.25.1-1ubuntu3.11) ...
Setting up liberror-perl (0.17029-1) ...
Setting up git-man (1:2.25.1-1ubuntu3.11) ...
Setting up git (1:2.25.1-1ubuntu3.11) ...
Processing triggers for man-db (2.9.1-1) ...
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws/src$ git clone -b kinetic-devel https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git
Cloning into 'turtlebot3_simulations'...
remote: Enumerating objects: 3160, done.
remote: Counting objects: 100% (958/958), done.
remote: Compressing objects: 100% (177/177), done.
remote: Total 3160 (delta 833), reused 781 (delta 781), pack-reused 2202
Receiving objects: 100% (3160/3160), 15.40 MiB | 124.00 KiB/s, done.
Resolving deltas: 100% (1864/1864), done.
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws/src$ cd ..
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws$ catkin_make
Base path: /home/anjali/Desktop/workspace/ros_ws/turtlebot3_ws
Source space: /home/anjali/Desktop/workspace/ros_ws/turtlebot3_ws/src
Build space: /home/anjali/Desktop/workspace/ros_ws/turtlebot3_ws/build
Devel space: /home/anjali/Desktop/workspace/ros_ws/turtlebot3_ws/devel
Install space: /home/anjali/Desktop/workspace/ros_ws/turtlebot3_ws/install
Creating symlink "/home/anjali/Desktop/workspace/ros_ws/turtlebot3_ws/src/CMakeLists.txt" pointing to "/opt/ros/noetic/share/catkin/cmake/toplevel.cmake"
####
#### Running command: "cmake /home/anjali/Desktop/workspace/ros_ws/turtlebot3_ws/src -DCATKIN_DEVEL_PREFIX=/home/anjali/Desktop/workspace/ros_ws/turtlebot3_ws/devel -DCMAKE_INSTALL_PREFIX=/home/anjali/Desktop/workspace/ros_ws/turtlebot3_ws/install -G Unix Makefiles" in "/home/anjali/Desktop/workspace/ros_ws/turtlebot3_ws/build"
####
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
```



```
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws$ ls
build  devel  src
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws$ source devel/setup.bash
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws$ 
```

- Now save the model and world file created in the previous question inside the turtlebot3_gazebo folder at their respective places.
- Now create a launch file for the model and save it in model folder.
- Now run the following command on terminal:

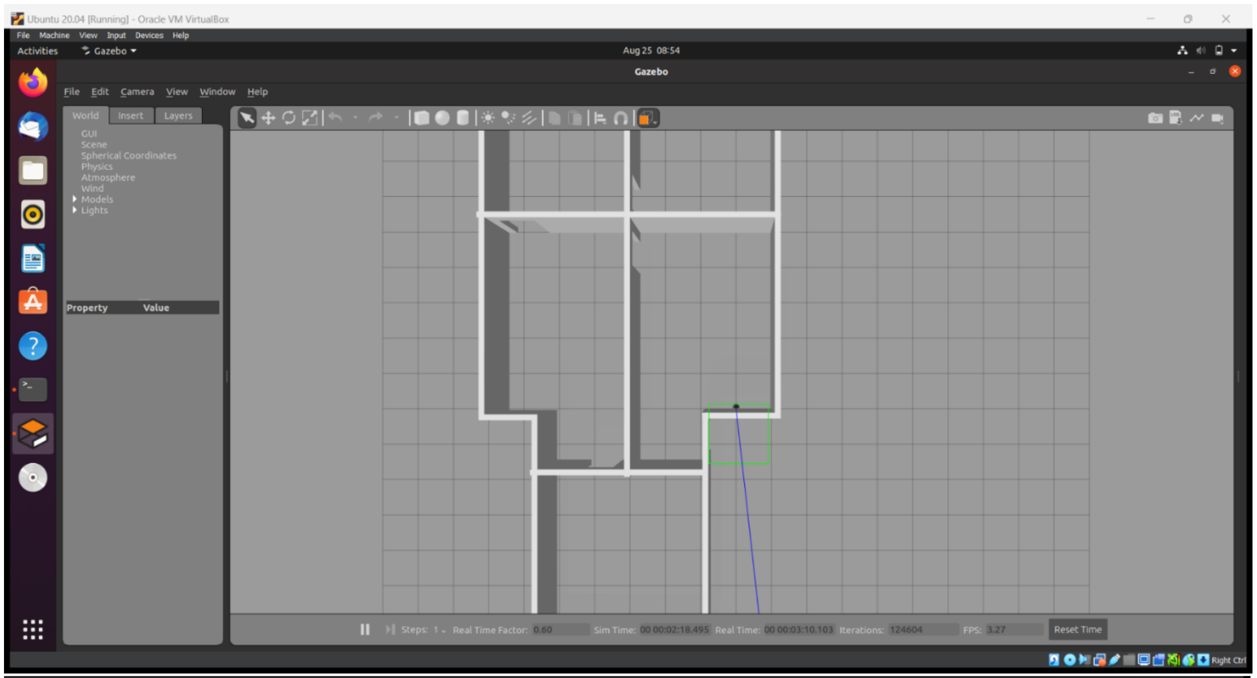
roslaunch turtlebot3_gazebo room_setup.launch

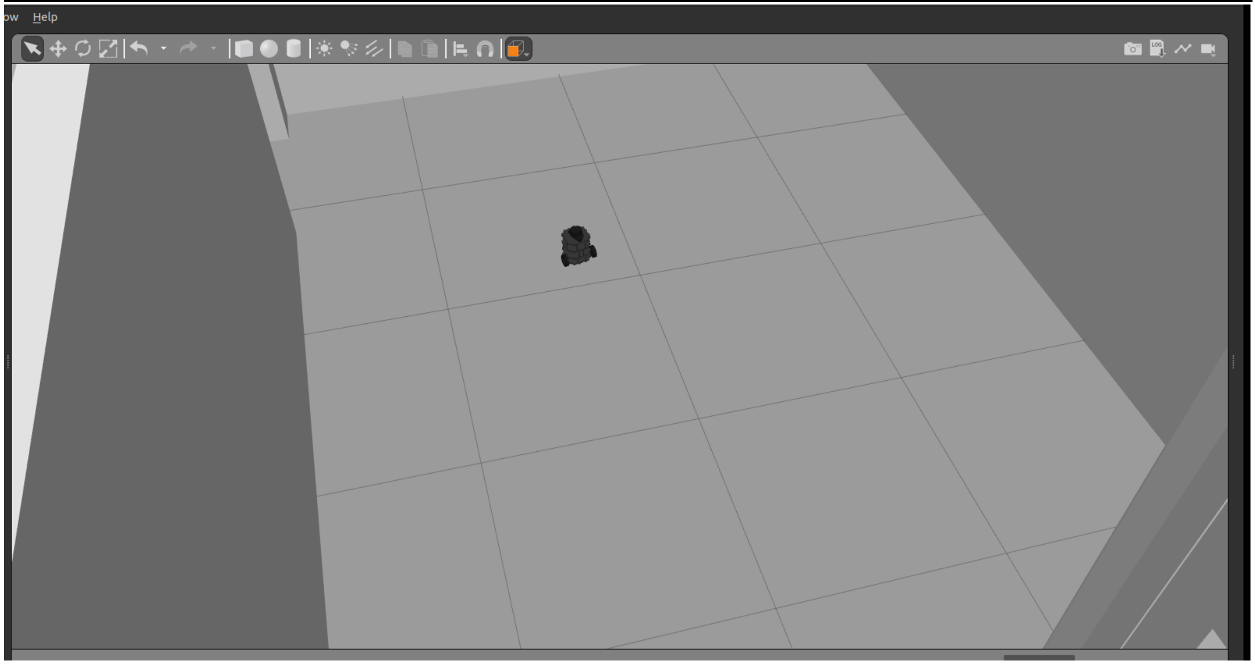


/home/anjali/Desktop/workspace/ros_ws/turtlebot3_ws/src/turtlebot3_simulations/turtlebot3_gazebo/launc...

```
anjali@anjali-VirtualBox:~$ cd Desktop/workspace/ros_ws/turtlebot3_ws
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws$ source devel/setup.bash
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws$ export TURTLEBOT3_MODEL=burger
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws$ roslaunch turtlebot3_gazebo room_setup.launch
... logging to /home/anjali/.ros/log/79485128-42f6-11ee-b94b-75f2a5a06bb5/roslaunch-anjali-VirtualBox-2432.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

xacro: in-order processing became default in ROS Melodic. You can drop the option.
started roslaunch server http://anjali-VirtualBox:32827/

SUMMARY
========

PARAMETERS
 * /gazebo/enable_ros_network: True
 * /robot_description: <?xml version="1....
 * /rosdistro: noetic
 * /rosversion: 1.16.0
 * /use_sim_time: True

NODES
  /
    gazebo (gazebo_ros/gzserver)
    gazebo_gui (gazebo_ros/gzclient)
```

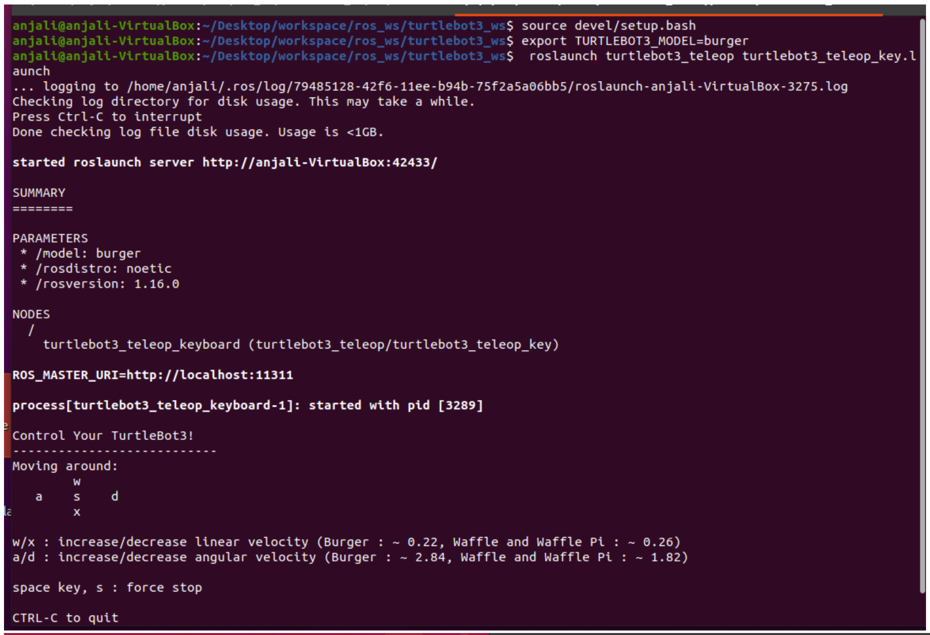
- The above command will run the environment with turtlebot3 with the room_setup environment we have created.


Ubuntu 20.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Gazebo
Aug 25 08:54
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property Value
Steps: 1
Real Time Factor: 0.60
Sim Time: 00 00:02:18.495
Real Time: 00 00:03:10.103
Iterations: 124604
FPS: 3.27
Reset Time
Right Ctrl

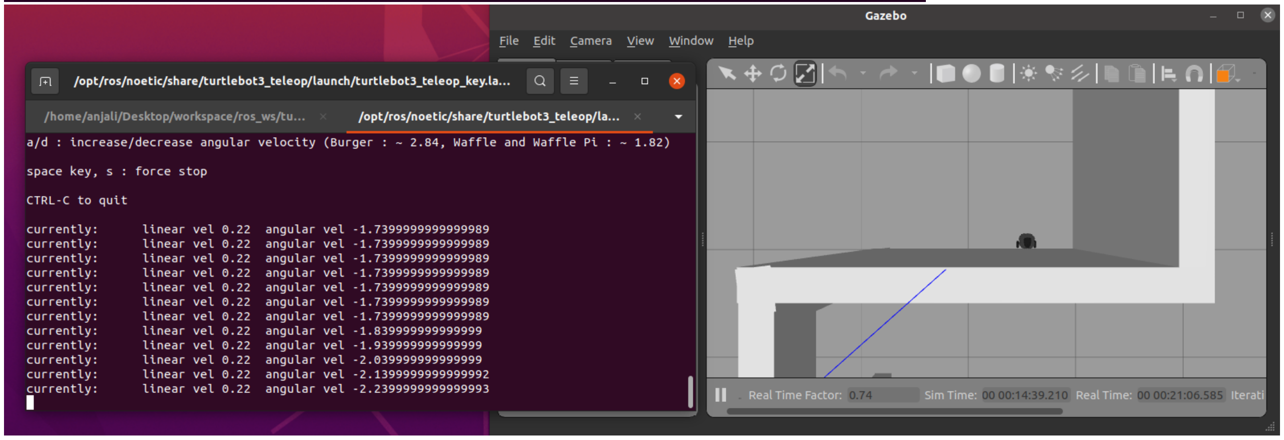

ow Help

- Now, to run the turtlebot3 using the keyboard run the following command in a new terminal:

roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch

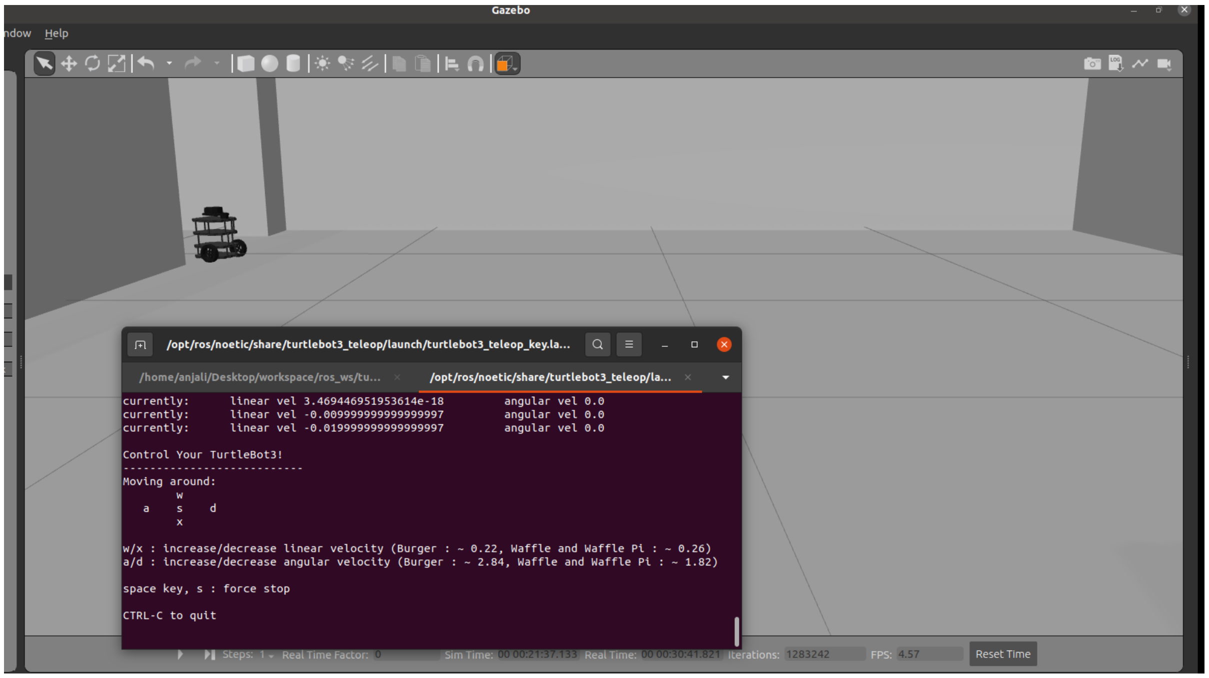


```
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws$ source devel/setup.bash
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws$ export TURTLEBOT3_MODEL=burger
anjali@anjali-VirtualBox:~/Desktop/workspace/ros_ws/turtlebot3_ws$  roslaunch turtlebot3_teleop turtlebot3_teleop_key.l
aunch
... logging to /home/anjali/.ros/log/79485128-42f6-11ee-b94b-75f2a5a06bb5/roslaunch-anjali-VirtualBox-3275.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://anjali-VirtualBox:42433/

SUMMARY
========

PARAMETERS
 * /model: burger
 * /rosdistro: noetic
 * /rosversion: 1.16.0

NODES
  /
    turtlebot3_teleop_keyboard (turtlebot3_teleop/turtlebot3_teleop_key)

ROS_MASTER_URI=http://localhost:11311

process[turtlebot3_teleop_keyboard-1]: started with pid [3289]

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
```



/opt/ros/noetic/share/turtlebot3_teleop/launch/turtlebot3_teleop_key.la...
/home/anjali/Desktop/workspace/ros_ws/tu...
/opt/ros/noetic/share/turtlebot3_teleop/la...
Gazebo
File
Edit
Camera
View
Window
Help
Real Time Factor: 0.74
Sim Time: 00 00:14:39.210
Real Time: 00 00:21:06.585
Iterati

```
a/d : increase/decrease angular velocity (Burger : ~ 2.84, Waffle and Waffle Pi : ~ 1.82)

space key, s : force stop

CTRL-C to quit

currently:      linear vel 0.22  angular vel -1.7399999999999989
currently:      linear vel 0.22  angular vel -1.7399999999999989
currently:      linear vel 0.22  angular vel -1.7399999999999989
currently:      linear vel 0.22  angular vel -1.7399999999999989
currently:      linear vel 0.22  angular vel -1.7399999999999989
currently:      linear vel 0.22  angular vel -1.7399999999999989
currently:      linear vel 0.22  angular vel -1.7399999999999989
currently:      linear vel 0.22  angular vel -1.839999999999999
currently:      linear vel 0.22  angular vel -1.939999999999999
currently:      linear vel 0.22  angular vel -2.039999999999999
currently:      linear vel 0.22  angular vel -2.1399999999999992
currently:      linear vel 0.22  angular vel -2.2399999999999993
```


Gazebo
ndow
Help
/opt/ros/noetic/share/turtlebot3_teleop/launch/turtlebot3_teleop_key.la...
/home/anjali/Desktop/workspace/ros_ws/tu...
/opt/ros/noetic/share/turtlebot3_teleop/la...
Steps: 1
Real Time Factor: 0
Sim Time: 00 00:21:37.133
Real Time: 00 00:30:41.821
Iterations: 1283242
FPS: 4.57
Reset Time

```
currently:      linear vel 3.469446951953614e-18         angular vel 0.0
currently:      linear vel -0.009999999999999997         angular vel 0.0
currently:      linear vel -0.019999999999999997         angular vel 0.0

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
```

Observation: Through gazebo we can build our own environment and run the robot of our choice in it.