
Industria Elektronikaren eta Automatikaren Ingeniaritzako Gradua

2. ERANSKINA:

ROS 2ren OINARRIZKO GIDA I

ROS 2ren BIDEZ LAU GURPIL FINKOKO
ROBOT BATEN NABIGAZIO
AUTONOMORAKO KONFIGURAZIOA

Ikaslearen izen eta abizenak: Iñaki Azkarragaurizar Jauregi

Irakaslearen izen eta abizenak: Mikel Larrea Sukia

Data: Donostia, 2023ko uztailaren 21a



Aurkibidea:

Argazkien Zerrenda	4
Komandoen Zerrenda	5
Programazio-Kodeen Zerrenda	6
Akronimoen Zerrenda	7
1. SARRERA	8
1.1 Aurrebaldintzak	8
1.2 Robota	9
2. LAN-ESPAZIOA ETA PAKETEA SORTU	10
3. ROBOT MUGIKORRA SORTU	11
3.1 URDF	11
3.2 RViz	13
3.3 Launch file:	14
3.4 Dependentziak (package.xml)	15
3.5 CMakeLists.txt	15
3.6 Paketea konpilatu	15
3.6 Jaurti robota RViz-en	17
3.7 rqt_graph	18
4. GAZEBO	19
4.1 Sortu mundua	19
4.2 Launch file	20
4.3 Martxan jarri	21
4.5 Nodoen zerrenda:	22
4.6 Nodoen informazioa	22
4.7 Gaien zerrenda	23
4.8 Gaien informazioa	24
4.9 Robota kontrolatu	24
4.10 Rqt_graph	27
5. NABEGAZIOA	28
5.1 Lokalizazioa	28
5.2 Mapaketa	32
6. BIBLIOGRAFIA	42
7. ERANSKINA: PROGRAMAZIO-KODEAK	43

Argazkien Zerrenda

1. Irudia: Junki_bot robota.	8
2. Irudia: proiektuaren hasierako egitura.	9
3. Irudia: proiektuaren egitura orain arte.	15
4. Irudia: lau gurpildun robota RViz-en.	16
5. Irudia: robotaren koordenatu-markoaren zuhaitza.	17
6. Irudia: nodoen komunikazioa grafikoki.	17
7. Irudia: Gazebo simulazio-ingurunea.	18
8. Irudia: Junki_bot-a Gazebo simulazio-ingurunean.	20
9. Irudia: Junki_bot-aren posizio berria.	24
10. Irudia: nodoen konexioen irudikapena.	26
11. Irudia: Junki_bot-a LIDAR sentsorearekin.	28
12. Irudia: LaserScan aktibatu RVizen.	30
13. Irudia: oztopoen detekzioa RVizen, lidarrarari esker.	31
14. Irudia: junki_bot-a SLAMarekin.	34
15. Irudia: junki_bot-a SLAMarekin, mapa estatikoa sortzen.	35
16. Irudia: lan-espazioaren egitura.	36
17. Irudia: mapa estatikoa kargatuta RVizen.	38
18. Irudia: AMCL aktibatuta.	38
19. Irudia: nabegazioa AMCLrekin.	39
20. Irudia: transformatuen zuhaitza.	40

Komandoen Zerrenda

1. Komandoa: karpeta bat sortzeko komandoa.	9
2. Komandoa: C++-eko paketea sortzeko komandoa.	9
3. Komandoa: Fitxategi bat sortzeko komandoa.	10
4. Komandoa: konpilatzeko komandoa.	14
5. Komandoa: lan-espazioa kargatzeko komandoa.	15
6. Komandoa: “launch” fitxategi bat abiarazteko komandoa.	16
7. Komandoa: exekutagarri bat abiarazteko komandoa.	16
8. Komandoa: nodoen konexioak era grafiko batean ikusteko komandoa..	17
9. Komandoa: gazebo abiarazteko komandoa.	18
10. Komandoa: aktibo dauden nodoen zerrenda bat ikusteko komandoa.	21
11. Komandoa: nodo baten informazioa ikusteko komandoa.	21
12. Komandoa: aktibo dauden gaien zerrenda bat ikusteko komandoa.	22
13. Komandoa: gai baten informazioa ikusteko komandoa.	23
14. Komandoa: mapa estatiko bat sortzeko komandoa.	35

Programazio-Kodeen Zerrenda

1. Programazio-kodea: deskribapena.xacro	44
2. Programazio-kodea: gazebo_kontrol.xacro	45
3. Programazio-kodea: junki_bot.urdf.xacro	45
4. Programazio-kodea: rviz_robota.launch.py	47
5. Programazio-kodea: urdf_config.rviz	48
6. Programazio-kodea: package.xml	49
7. Programazio-kodea: CMakeLists.txt	49
8. Programazio-kodea: etxea.world	65
9. Programazio-kodea: gazebo.launch.py	67
10. Programazio-kodea: lidar.xacro	69
11. Programazio-kodea: nav2_params.yaml	74
12. Programazio-kodea: nav2_default_view.rviz	84
13. Programazio-kodea: rviz_launch.py	86
14. Programazio-kodea: nabegazioa.launch.py	88

Akronimoen Zerrenda

GIE Gipuzkoako Ingeniaritza Eskola

GrAL Gradu Amaierako Lana

ROS Robot Operating System

XML eXtensible Markup Language

EOL End of Life

URDF Unified Robot Description Format

SDF Simulation Description Format

SLAM Simultaneous Localization And Mapping

AMCL Adaptive Monte Carlo Localization

LiDAR Light Detection and Ranging

1. SARRERA

Tutorial honek urratsez urrats azalduko du nola konfiguratu nabegazioa robot batean, ROS 2 Humble Hawksbill bertsioa erabiliz. Eraikiko den robota Junki_bot izena izango du eta lau gurpil izango ditu, non hauetako bakoitza finkoa eta trakzioduna izango da, beraz, horietako bat ere ez da angelu batean biratuko.

1.1 Aurrebaldintzak

- Robot Operating System 2ren Oinarrizko Gida I, osorik osatua.
- Gida honek The Construct [1] plataforma erabiltzen du, robotentzako simulazio-ingurune bat eskaintzen baitu. Horri esker, gida honetan erabiliko diren pakete asko deskargatuta egongo dira. Bertan, soilik, eskuragarri dauden paketeen zerrenda egunerautu eta sisteman instalatutako paketeak egunerautu beharko dira. Horretarako:

Terminal bat ireki:

```
sudo apt update
```

Komando hori exekutatzean, sistemak instalatutako paketeetarako egunerautze erabilgarriak bilatuko ditu, eta tokiko biltegiak informazioa egunerautuko du.

Terminal berdinean:

```
sudo apt-get upgrade
```

Sisteman instalatutako paketeak egunerautzeko erabiltzen da komando hau. Horri esker, plataforma honetan nabegazioko liburutegiak, beste zenbaiten artean erabiltzea ahalbidetuko da.

The Construct platformari buruzko informazio gehiago, bere web orrian aurki daiteke [1].

- ROS 2 Humble Hawksbill [2] bertsioa erabiliko da, eta horretarako The Construct plataforma erabiltzea gomendatzen da. Baino horrela ez bada, ROS 2n [2] eta Navigation 2ren [3] webgune ofizialean behar den guztia deskarga daiteke.

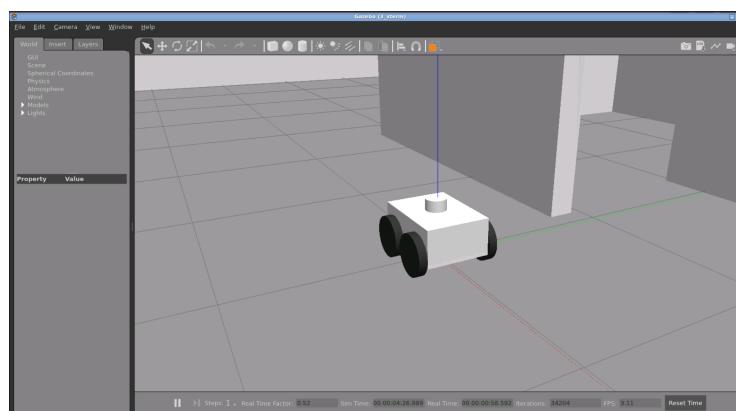
1.2 Robota

Junki_bot-a, trakzio finkoko lau gurpileko robot bat da. Eta gida honen helburua, Gazeboko simulazio-ingurune batean autonomiaz nabigatzea da, sentsoreetatik bildutako informazioari esker. Hori hurrengo urratsetan egingo da:

1. Robota zerotik sortu Xacro fitxategien bidez eta behar diren konfigurazio guztiak konfiguratu, RVizen bistaratzeko.
2. Robota Gazebo simulazio-eremuan abiatu, honen mugimendua kontrolatzen duen nodoarekin.
3. Robotari nabegaziao gehitu, horretarako, beharrezko sentsore, pakete eta parametroak konfiguratu.

Sortuko den robota, Junki_bot izena izango du eta ondorengo osagai fisikoak izango ditu:

1. Txasisa (zuria).
2. Lau trakzio-gurpil finko (beltzak).
3. LIDAR sentsorea (zuria).



1. irudia: junki_bot robota.

1. irudiko robota sortzeko eta ingurune simulatu batean bakarrik nabigatzeko, jarraitu gida honek eskaintzen dituen atal guztiak pausoaz pauso.

2. LAN-ESPAZIOA ETA PAKETEA SORTU

Robotarekin lanean hasi baino lehen, lehenik eta behin lan-espazio bat sortuko da, hau da, proiektu honen karpeta izango dena. Horretarako terminal bat ireki:

```
mkdir -p junki_ws/src
```

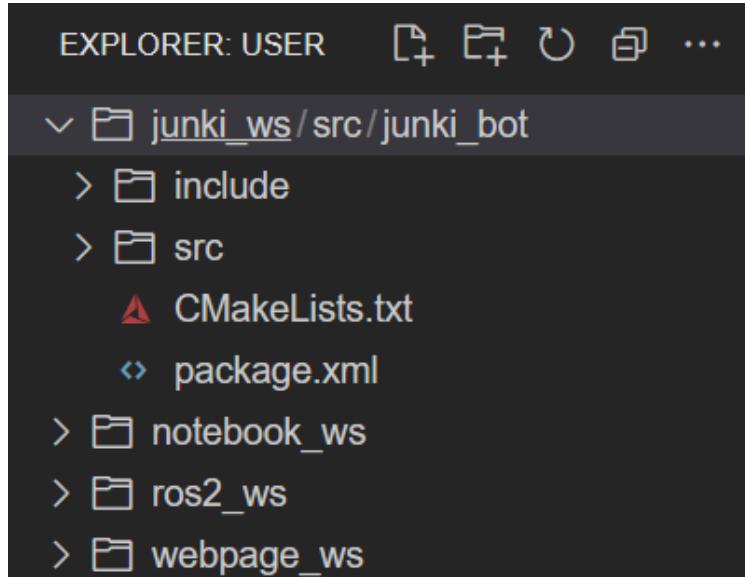
1. Komandoa: karpeta bat sortzeko komandoa.

Ondoren, junki_bot izeneko C++ motako paketea eraikiko da aurretik sortutako lan-espazioaren barruko src karpetan, horretarako terminal berdinean:

```
cd junki_ws/src/  
ros2 pkg create --build-type ament_cmake junki_bot
```

2. Komandoa: C++-eko paketea sortzeko komandoa.

“Code Editor”-ean sartuta, 2. irudian ikus daitekeen egitura berdina eduki beharko litzateke.



2. Irudia: proiektuaren hasierako egitura.

3. ROBOT MUGIKORRA SORTU

3.1 URDF

Robot mugikor simulatu bat eraikitzeko, ROS 2k, Universal Robot Description Format (URDF) erabiltzen du. Bertan, robotaren propietate fisiko guztiak deskribatuko dira. Fitxategi mota horrek XML lengoia mota gisa erabiltzen du.

Hain handia izan ez dadin, ROSEk Xacro izeneko pakete bat dauka. Pakete honek robot-ereduak edo osagai robotikoak modu modularragoan deskribatzeko aukera ematen du. URDF fitxategiak bezala, XML motako lengoia erabiltzen du.

Fitxategi hauek, deskribapena izeneko karpeta baten barruan joango dira. Horretarako, terminal bat ireki:

```
cd ~/junki_ws/src/junki_bot/  
mkdir deskribapena
```

Oraintxe sortutako karpetaren barruan, lehenengo fitxategia sortuko da, non Junki_bot robotaren egitura fisikoak deskribatuko dira, XML formatua erabiliz. Terminal batean:

```
cd deskribapena/  
touch deskribapena.xacro
```

3. Komandoa: Fitxategi bat sortzeko komandoa.

Egin klik [deskribapena.xacro](#) atalean kode osoa ikusteko.

Kodearen azalpena:

1. Balio konstanteak: Hasieran zenbait propietate definitzen dira, hala nola robotaren onarriarekin eta gurpilekin lotutako dimentsio era parametroak.
2. Propietateak: Kutxa eta zilindro baten propietate inertzialak zehazten dira, aurrerago robotaren txasia eta gurpilak sortzeko erabiliko dira.

3. Base_link: Lotura hau, robotaren oinarria izango da. Kutxa baten itxurakoa da eta bere propietate inertzialak ditu.
4. Base_footprint: Lotura honek, robotak lurrean duen proiekzioa irudikatzen du. Ez du inolako geometriarik, erreferentzia puntu gisa erabiltzen da beste lotura batzuekiko.
5. Base_joint: “Fixed” motatako artikulazio honek, base_footprint eta base_link loturak konektatzen ditu. Hierarkia mota baten, base_footprint bestearren gainetik jarritakoa.
6. Makro Atzeko gurpilak: Makro honen bitartez atzoko ezker eta eskuin gurpilen loturak eta artikulazioak definitzen dira. Gurpil bakoitza, “continuous” artikulazio mota du, etengabe biratu ahal izateko. Eta base_link-erako lotura dute.
7. Makro Aurreko gurpilak: seigarren puntuaz azaldu den berdina da, baina aurreko gurpilen kasurako.

Orain, karpeta berdinean, gazebo_kontrola.xacro izeneko fitxategia sortuko da. Honek, Gazebo eta ROS pluginak eratzen ditu, robotaren mugimendua kontrolatzea ahalbidetuz.

Terminal batean:

```
cd deskribapena/  
touch gazebo_kontrola.xacro
```

Egin klik [gazebo_kontrol.xacro](#) atalean kode osoa ikusteko.

Kodearen azalpena:

1. Marruskadurak: Hasieran gurpil lotura bakoitzari, marruskadura estatiko eta dinamiko balore bat ezartzen zaio. Gainera, gurpilak kolore grisa esleitzen zaie gazebo mundurako.
2. libgazebo_ros_diff_drive.so: Plugin hau, trakzio diferenzialeko robotaren mugimendua kontrolatzeko erabiltzen da. Gainera, gurpilekin lotutako zenbait parametro ezartzen dira, hala nola gurpilen diametroa eta abar. Horretaz gain, odometria datuak argitaratuko dira, aurrerago robota nabegatzeko oso garrantzitsua izango dena.

3. libgazebo_ros_joint_state_publisher.so: Plugin hau, gurpilen artikulazioaren egoeraren informazioa argitalatzeko baliagarria izango da. Oso baliotsua izango da nabegazio atalerako.

Amaitzeko, Junki_bot izeneko fitxategi goren bat sortu da, eta hortik Xacro fitxategi guztiak abiaraziko dira. Hau, kodea garbiagoa eta egituratuagoa izan dadin egiten da. Terminal bat ireki:

```
cd deskribapena/  
touch junki_bot.urdf.xacro
```

Egin klik [junki_bot.urdf.xacro](#) atalean kode osoa ikusteko.

URDF fitxategiei buruzko informazio zabalagoa eta zehatzagoa izateko dokumentazio ofizialean aurki daiteke [4].

3.2 RViz

RViz oso erabilgarria da datu sentsorialak, robotaren transformatuak eta abar ikusteko. Beraz, oso baliotsua izango da robotak sortutako datuak bistaratzeko eta aztertzeko.

Terminal batean, bere karpeta propioa eta barruan urdf_config.rviz fitxategia sortu.

```
cd ~/junki_ws/src/junki_bot/  
mkdir rviz  
cd rviz/  
touch urdf_config.rviz
```

Egin klik [urdf_config.rviz](#) atalean kode osoa ikusteko.

Fitxategi hau, RViz bistaratze konfigurazioa deskribatzeko erabiltzen da. Honi esker, panelak, bistaratzeak, tresnak eta konfigurazio globalen itxurak konfiguratu daiteke.

RVizi eta bere funtzionaltasunari buruzko informazio gehiago lortzeko, kontsultatu dokumentazio ofizialean [5]

3.3 Launch file:

Jauritzeko fitxategia edo ezagunago “Launch file”, sortutako robota eta beharrezko nodoak kargatzeko erabiliko da. Terminal bat ireki, eta karpeta berri bat sortuko da eta bere barruan dagokion jaurtitze fitxategia:

```
cd ~/junki_ws/src/junki_bot/  
mkdir launch  
cd launch/  
touch rviz_robota.launch.py
```

Egin klik [rviz_robota.launch.py](#) atalean kode osoa ikusteko.

Kodearen azalpena:

1. Paketeen importaketa: Fitxategi hasieran, beharrezko moduluak eta paketeak importatzen dira, “from” eta “import” izeneko komandoen bitartez.
2. Karpeten direkzioa: Robotak behar dituen fitxategi eta karpeten direkzioa adierazten da, lehenengo zatian zehaztutako “os” eta “FindPackageShare” paketeen bidez.
3. Aldagaiak ezarri: “LaunchConfiguration” paketearen bitartez, aldagaiak ezarri dira. Honek, robotaren simulazioa martxan jartzea ahalbidetuko du.
4. start_joint_state_publisher_cmd: Nodo honek, robotaren URDF fitxategian finkoak ez diren artikulazioen egoera-balioak argitaratzeko ardura du.
5. start_joint_state_publisher_gui_node: Aurrekoaren antzekoa da, baina egoera-balioak manipultzeko interfaze bat eskaintzen du.
6. start_robot_state_publisher_cmd: Nodo honi esker, robotaren artikulazio egoeretara harpidetzen da eta lotura bakoitzaren orientazioa argitaratzen du.
7. start_rviz_cmd: Nodo honi esker, RViz aplikazioa abiarazten du.
8. return: Azkenik, aurretik deklaratutako ekintza guztiak “LaunchDescription” objetuari gehitzen zaizkio, eta emaitza gisa itzultzen da.

Fitxategi hauei buruzko informazio gehiago lortzeko, konsulatatu ROS 2ren dokumentazio ofizialean [2].

3.4 Dependentziak (package.xml)

“Package.xml” fitxategiak, paketeari buruzko informazioa ematen du, hala nola izena, proiektuaren deskribapena, egileak, dependentziak eta erlazionatutako beste konfigurazio batzuk. Proiektuaren mende egongo diren pakete batzuk gehitzea beharrezko da, horretarako fitxategi hori ireki eta gehitu hurrengoa:

```
<exec_depend>joint_state_publisher</exec_depend>
<exec_depend>robot_state_publisher</exec_depend>
<exec_depend>rviz2</exec_depend>
<exec_depend>xacro</exec_depend>
```

Egin klik [package.xml](#) atalean kode osoa ikusteko.

3.5 CMakeLists.txt

Paketea konpilatu ahal izateko, beharrezko konpilazio-konfigurazioa ezartzeaz eta iturburuko dependentzia eta artxiboei buruzko informazioa emateaz arduratzen da. Orain arte sortutako karpeta eta fitxategiak sisteman instalatu ahal izateko, hurrengo kode zatia gehitu CMakeLists.txt fitxategian:

```
install(
  DIRECTORY src launch rviz deskribapena
  DESTINATION share/${PROJECT_NAME}
)
```

Hau, if(BUILD_TESTING) kode-lerroaren aurretik jartzea gomendatzen da. Egin klik [CMakeLists.txt](#) atalean kode osoa ikusteko.

3.6 Paketea konpilatu

Puntu honetan, sortutako paketea arazorik gabe konpilatu ahal da. Akzio hau, lan-espazioaren barruan egin behar da beti, hau oso garrantzitsu da. Horretarako, terminal bat ireki:

```
cd ~/junki_ws/
colcon build
```

4. Komandoa: konpilatzeko komandoa.

Gogoratu: Lan-espazioan aldaketa bat egiten den bakoitzean, egin beharreko pausu bat da eta beti lan-espazioaren barruan exekutatu behar da akzio hau.

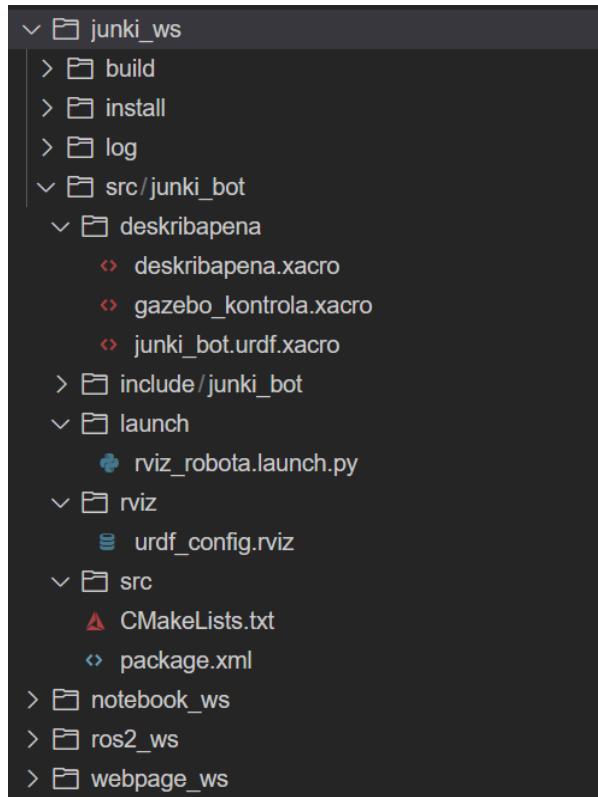
Lan-ingurunean instalatutako paketeak ezagutu eta eskuratu ahal izateko, komando hau kopiatu:

```
cd ~/junki_ws/  
source install/setup.bash
```

5. Komandoa: lan-espazioa kargatzeko komandoa.

Komando hau ere, lan-espazioaren barruan exekutatu behar da, proiektua erabili behar den terminal bakoitzean.

Orain arte egindakoarekin, “Code Editor”-era joanda, 3. irudian agertzen den egitura eduki beharko litzateke.



3. Irudia: proiektuaren egitura orain arte.

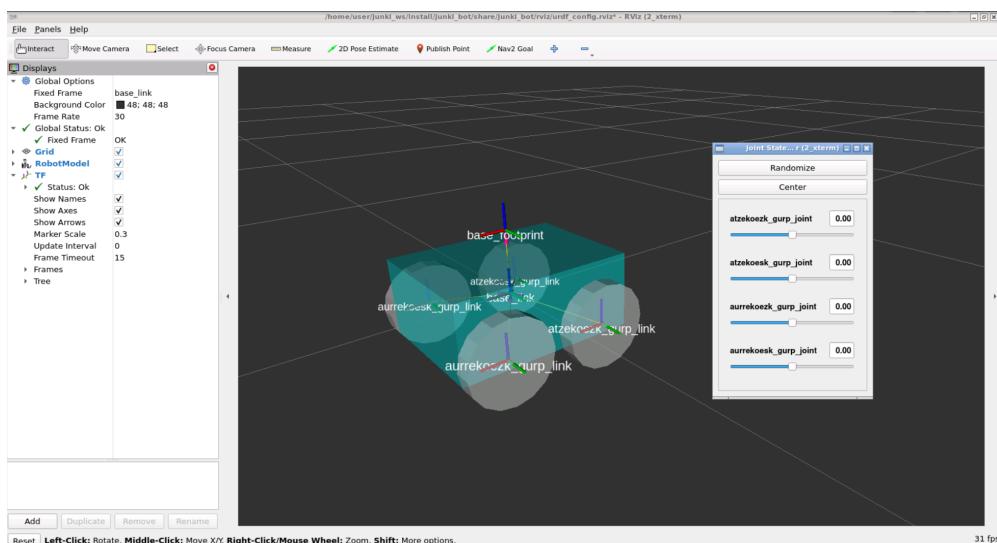
3.6 Jaurti robota RViz-en

Aurreko pausu guztiak jarraitu ostean, 3.5 ataleko terminal berdinean exekutatu “rviz_robota.launch.py” fitxategia:

```
ros2 launch junki_bot rviz_robota.launch.py
```

6. Komandoa: “launch” fitxategi bat abiarazteko komandoa.

Behin exekutatuta, RViz abiaratu beharko litzateke sortutako lau gurpildun robotarekin, 4. irudian ikus daitekeen bezala.



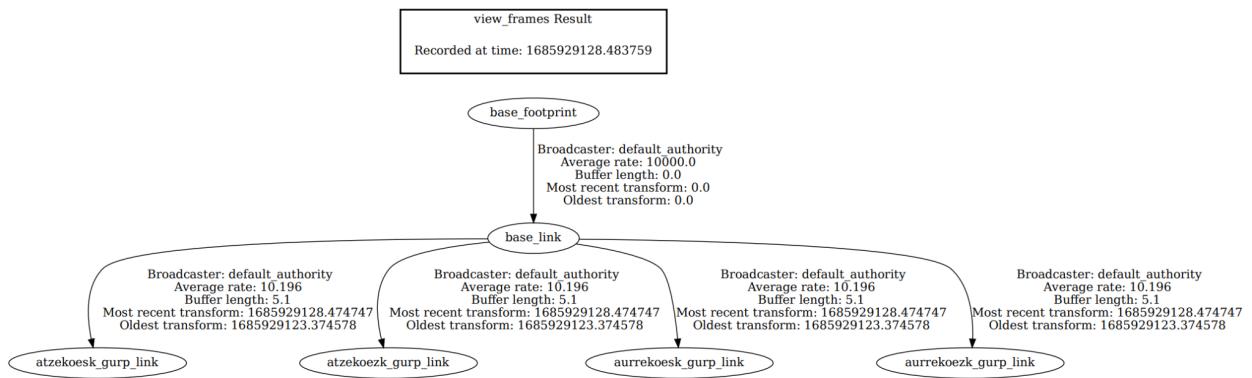
4. Irudia: lau gurpildun robota RViz-en.

Robotarekin batera, leihoa bat ireki da. Hau, “start_joint_state_publisher_gui_node” izeneko nodo bat da eta honi esker gurpil bakoitzaren zinematika ikustea ahalbidetuko du.

Sortutako ibilgailuaren koordenatu-markoak ikusi ahal izateko, terminal berri bat ireki:

```
ros2 run tf2_tools view_frames
```

7. Komandoa: exekutagarri bat abiarazteko komandoa.



5. Irudia: robotaren koordenatu-markoaren zuhaitza.

Aurretik aipatu den bezala, loturak modu hierarkiko batean egiten dira, base_footprinta goian izanik eta lau gurpilen loturak zuhitzaren azpian geratuz.

3.7 rqt_graph

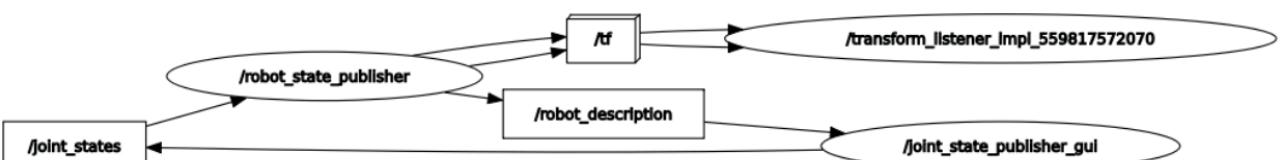
Tresna honen bitartez, aktibo dauden nodoen komunikazioa ikustea ahalbidetzen du.

Terminal bat ireki:

rqt_graph

8. Komandoa: nodoen konexioak era grafiko batean ikusteko komandoa..

Komando hori exekutatu ostean, ireki berri den lehioaren goikaldean “Node/Topics (all)” aktibatu. Hori egin ondoren, nodo eta gaien komunikazioa era bisual batean ikusi beharko litzatek, 6. irudian ageri den bezala.



6. Irudia: nodoen komunikazioa grafikoki.

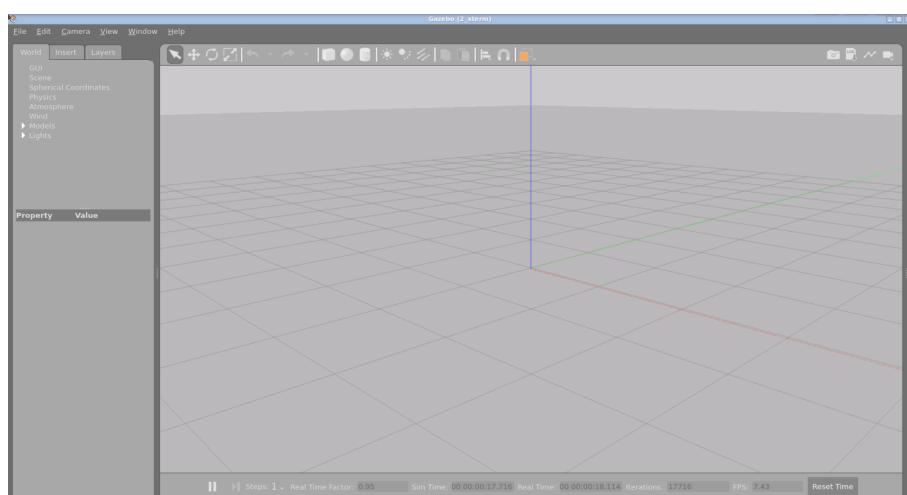
Goiko irudi honetan ikusi daiteke nola “/joint_state_publisher_gui” nodoak “/joint_state” gaiean datuak (gurpilaren angeluak) bidaltzen dituen. Datu horiek, “/robot_state_publisher” nodoak jasotzen ditu eta koordenatu-marko horien informazioa “tfn” argitaratzen ditu. Eta horrela, RVizen roboteko gurpilen posizio berria eguneratzen da.

4. GAZEBO

Gazebo, ROS 2ren barruan implementatutako aplikazio bat da, simulazio-ingurune bezala jokatzen duena. Honi esker, Junki_bot-aren algoritmoak probatu ahalko dira 3 dimentsiotan, robotak izango duen portaera ikusiz. Gazebo abiarazteko, terminal bat ireki eta izen bera idatzi:

gazebo

9. Komandoa: gazebo abiarazteko komandoa.



7. Irudia: Gazebo simulazio-ingurunea.

7. irudian, Gazeboaren simulazio-ingurunea ikus daiteke. Momentuz hutsik dago, baina hurrengo ataletan, sortutako robota bertan abiaraziko da. Aplikazio honi buruzko informazio ofizial gehiago dokumentazio ofizialean aurki daiteke [6].

4.1 Sortu mundua

Etxe itxurako mundu bat eraikiko da, robota ingurune horretan ibili dezan. Karpeta espezifiko bat sortuko da, mundu desberdinak bertan gordetzeko eta bere barruan “etxea.world” izeneko fitxategia. Terminal bat ireki:

```
cd ~/junki_ws/src/junki_bot/  
mkdir worlds/  
cd worlds/  
touch etxea.world
```

Egin klik [etxea.world](#) atalean kode osoa ikusteko.

Kode hau, Gazebo simulagailuko eredu baten deskribapena da, SDF formatuan. Gida honetan, ez du garrantzi berezirik fitxategi hau azaltzea. Honi buruzko informazio gehiago, Gazeboko dokumentazio ofizialean aurki daiteke [6].

Gogoan izan, “worlds” izeneko karpeta berri bat sortu dela eta konpilatu ahal izateko, [CMakeList.txt](#) eguneratu behar dela. Beraz, fitxategi horretan “worlds” karpeta gehitu.

```
install(  
  DIRECTORY src launch rviz deskribapena worlds  
  DESTINATION share/${PROJECT_NAME}  
)
```

4.2 Launch file

Lau gurpildun ibilgailua eta etxe itxurako mundua Gazebon martxan jartzeko, “gazebo.launch.py” izeneko fitxategi bat sortuko da. Horretarako, “launch” izeneko direktoriora joan eta bertan sortu:

```
cd ~/junki_ws/src/junki_bot/launch/  
touch gazebo.launch.py
```

Egin klik [gazebo.launch.py](#) atalean kode osoa ikusteko.

Kodearen azalpena:

1. Fitxategi hasieran, beharrezko moduluak eta paketeak importatzen dira.
2. Gero, robotak behar dituen fitxategi eta karpeten direkzioa adierazten da.
3. Ondoren, “LaunchConfiguration” aldagaiak ezarri dira. Honek, robotaren simulazioa martxan jartzea ahalbidetuko du.
4. Ostean, “DeclareLaunchArgument” klasea erabiliz zenbait argudio deklaratzen dira. Klase hori, konfigurazio gehigarriak zehazteko erabiltzen da, adibidez: simulazio denbora; izenen espazioa; robotaren eredu eta abar.
5. “ExecuteProcess” eta “Node” bezalako klaseak erabiliz, prozesu eta nodo desberdinak aktibatuko dira. Adibidez, Gazebo zerbitzaria, Gazebo bezeroa eta robotaren egoera argitalatzen duen nodoa aktibatzen dira.

6. Azkenik, aurretik deklaratutako ekintza guztiak “LaunchDescription” objetuari gehitzen zaizkio, eta emaitza gisa itzultzen da.

4.3 Martxan jarri

Oraintxe bertan sortutako “gazebo.launch.py” fitxategia abiarazteko, lehendabizi paketea konpilatu:

```
cd ~/junki_ws/  
colcon build
```

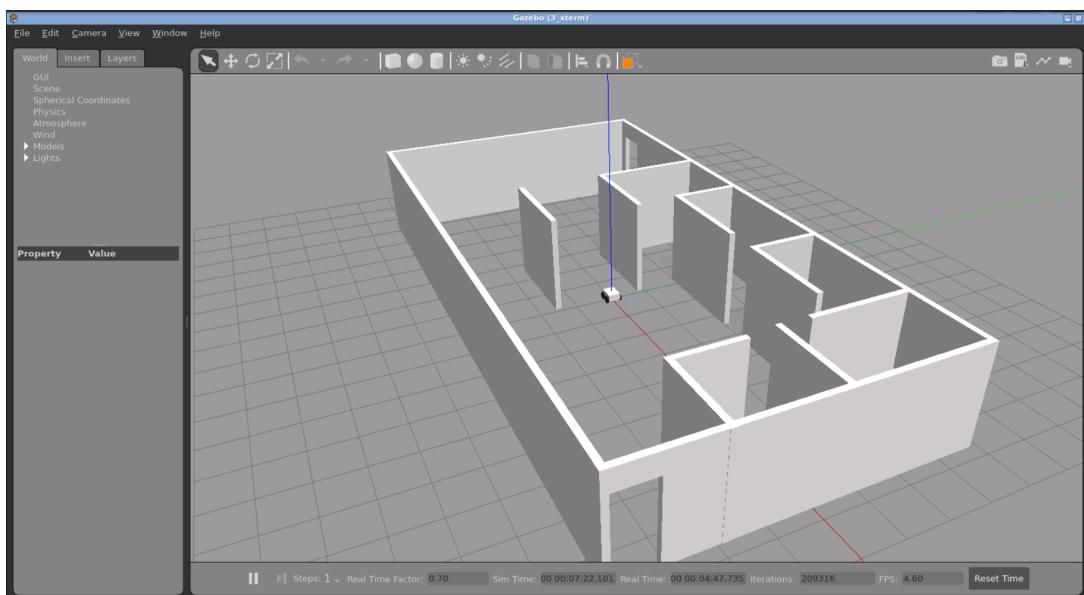
Paketeak lan-ingurunean instalatzeko:

```
cd ~/junki_ws/  
source install/setup.bash
```

Eta orain bai, fitxategia exekutatu:

```
ros2 launch junki_bot gazebo.launch.py
```

Behin hori exekutatuta, 8. irudian agertzen den bezala, lau gurpileko robota eta sortutako “world” fitxategia, gazebon ikusi beharko litzateke. Gogoratu gurpilak beltzak direla, horrela definitu direlako [gazebo_kontrol.xacro](#) fitxategian.



8. Irudia: Junki_bot-a Gazebo simulazio-ingurunean.

Robota, X=0, Y=0 eta Z=0 koordenatuau ageri da, horrela definitu delako [gazebo.launch.py](#) fitxategian. Abiapuntu bezala, beste leku batean nahi izan ezkero, aurreko koordenatuak aldatu beharko litzateje.

4.5 Nodoen zerrenda:

Aktibo dauden nodo guztien zerrenda bat ikusteko, terminal bat ireki:

```
ros2 node list
```

10. Komandoa: aktibo dauden nodoen zerrenda bat ikusteko komandoa.

Erantzuna:

```
/diff_drive  
/gazebo  
/junki_joint_state  
/robot_state_publisher
```

4.6 Nodoen informazioa

“Info” komandoari esker, aktibo dauden nodoen informazioa jaso daiteke. Adibidez, “/diff_drive-rena” ikusteko, terminal bat ireki:

```
ros2 node info /diff_drive
```

11. Komandoa: nodo baten informazioa ikusteko komandoa.

Erantzuna:

```
/diff_drive  
Subscribers:  
/clock: rosgraph_msgs/msg/Clock  
/cmd_vel: geometry_msgs/msg/Twist  
/parameter_events: rcl_interfaces/msg/ParameterEvent  
Publishers:  
/odom: nav_msgs/msg/Odometry  
/parameter_events: rcl_interfaces/msg/ParameterEvent  
/rosout: rcl_interfaces/msg/Log  
/tf: tf2_msgs/msg/TFMessage  
Service Servers:  
/diff_drive/describe_parameters: rcl_interfaces/srv/DescribeParameters  
/diff_drive/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
```

```
/diff_drive/get_parameters: rcl_interfaces/srv/GetParameters
/diff_drive/list_parameters: rcl_interfaces/srv/ListParameters
/diff_drive/set_parameters: rcl_interfaces/srv/SetParameters
/diff_drive/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
Service Clients:
```

Action Servers:

Action Clients:

Hemendik informazio ugari atera daiteke. Alde batetik, nodo hau zein gaietatik informazioa jasotzen duen, eta gainera informazio hori ze motatakoa den adierazten du. Adibidez, “/cmd_vel” gaira harpidetuta dago.

Beste alde batetik, nodo hau argitalatzaile bezala jokatzen duela adieratzen du. Adibidez, “/odom” izeneko gaiean robotaren posizioaren datuak bidaltzen ditu. Hau, aurrerago beharrezkoa izango da, ibilgailua bakarrik nabegatu dezan.

4.7 Gaien zerrenda

Aktibo dauden gai guztien zerrenda bat ikusteko, terminal bat ireki:

ros2 topic list

12. Komandoa: aktibo dauden gaien zerrenda bat ikusteko komandoa.

Erantzuna:

```
/clock
/cmd_vel
/joint_states
/odom
/parameter_events
/performance_metrics
/robot_description
/rosout
/tf
/tf_static
```

Zerrenda honetan, “/cmd_vel” izeneko gaia dago. Junki_bot ibilgailuari abiadura linealeko eta angeluarreko komandoak bidatzeko erabiltzen da.

Lista horretan, 4.6 atalean aipatutako “odom” gaia ageri da. Hau, robot mugikorraren odometria-datuak argitaratzeko erabiltzen da, aurretik [gazebo_kontrol.xacro](#) fitxategian ezarri da.

4.8 Gaien informazioa

Aktibo dauden gai guztiak informazioa ikusteko “info” komandoa erabiltzen da. Adibidez, “/cmd_vel” gaiarena ikusteko, terminal batean:

```
ros2 topic info /cmd_vel
```

13. Komandoa: gai baten informazioa ikusteko komandoa.

Erantzuna:

```
Type: geometry_msgs/msg/Twist
Publisher count: 0
Subscription count: 1
```

Lehenengo lerroak, gai horretan informazioa argitaratzeko edo jasotzea, zero motatako mezuak izan behar diren adierazten du. Ondoren, “topic-ak” harpidedun bat eta zero argitalatzaile duela esaten du. Ondorengo atalean, gai honetan informazioa argitaratuko duen nodo bat exekutatuko da, robota teklatuaren bitartez kontrolatuko duena.

4.9 Robota kontrolatu

Gaien informazioaren atalean ikusi den bezala, /cmd_vel “topic-a”, simulatutako ibilgailuei abiadura linealeko eta angeluarreko komandoak bidaltzeko erabiltzen da. Baino momentuz, ez du bertan argitaratzen duen nodorik, soluzio gisa terminal berri bat ireki eta hurrengo exekutagarria abiatu:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

Erantzuna:

This node takes keypresses from the keyboard and publishes them as Twist messages. It works best with a US keyboard layout.

Moving around:

u	i	o
j	k	l
m	,	.

For Holonomic mode (strafing), hold down the shift key:

U	I	O
J	K	L
M	<	>

t : up (+z)

b : down (-z)

anything else : stop

q/z : increase/decrease max speeds by 10%

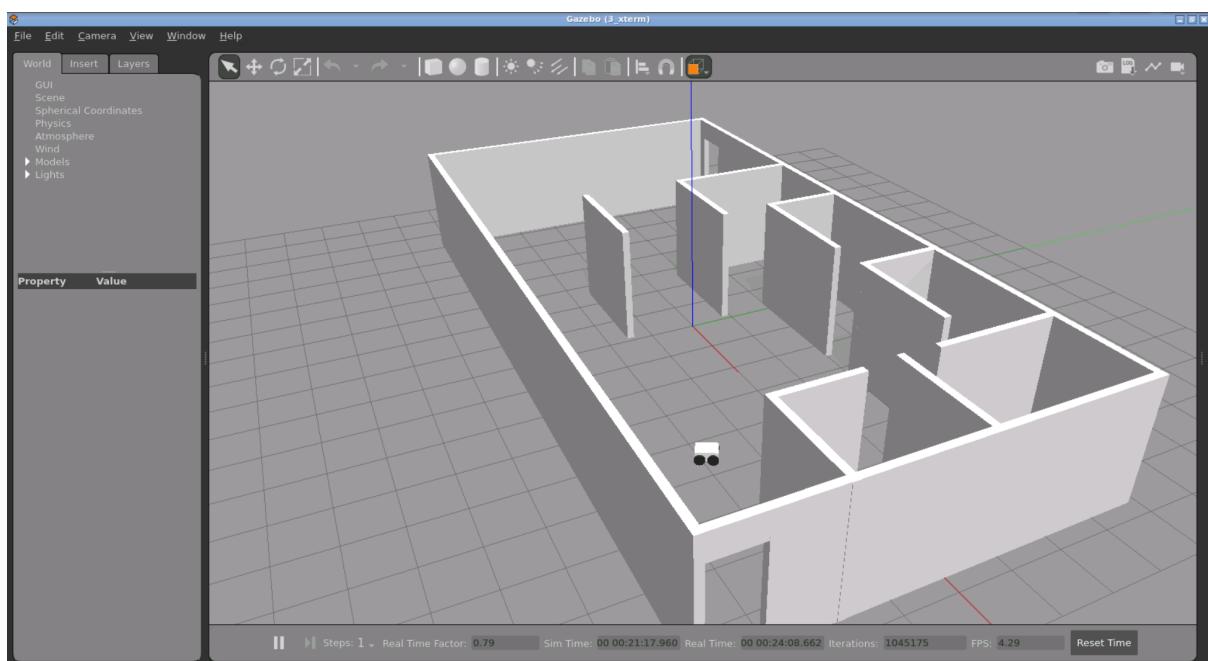
w/x : increase/decrease only linear speed by 10%

e/c : increase/decrease only angular speed by 10%

CTRL-C to quit

currently: speed 0.5 turn 1.0

Orain, teklatua erabiliz ibilgailua kontrolatu ahal izango da. 9. irudian ikus daiteke, nodo honi esker Junki_bot-a mugitu dela.



9. Irudia: Junki_bot-aren posizio berria.

Orain, aktibo dauden nodoen zerrendari erreparatuz, exekutatu berri den nodoa agertu beharko litzateke. Terminal bat ireki:

```
ros2 node list
```

Erantzuna:

```
/diff_drive  
/gazebo  
/junki_joint_state  
/robot_state_publisher  
/teleop_twist_keyboard
```

Beraz, ikusi daiteke /teleop_twist_keyboard nodoa zerrenda honetara gehitu dela. Nodo honi buruzko informazio gehiago ikusteko:

```
ros2 node info /teleop_twist_keyboard
```

Erantzuna:

```
/teleop_twist_keyboard
```

Subscribers:

Publishers:

```
/cmd_vel: geometry_msgs/msg/Twist  
/parameter_events: rcl_interfaces/msg/ParameterEvent  
/rosout: rcl_interfaces/msg/Log
```

Service Servers:

```
/teleop_twist_keyboard/describe_parameters: rcl_interfaces/srv/DescribeParameters  
/teleop_twist_keyboard/get_parameter_types: rcl_interfaces/srv/GetParameterTypes  
/teleop_twist_keyboard/get_parameters: rcl_interfaces/srv/GetParameters  
/teleop_twist_keyboard/list_parameters: rcl_interfaces/srv/ListParameters  
/teleop_twist_keyboard/set_parameters: rcl_interfaces/srv/SetParameters  
/teleop_twist_keyboard/set_parameters_atomically:  
rcl_interfaces/srv/SetParametersAtomically
```

Service Clients:

Action Servers:

Action Clients:

Nodo honek, ez du daturik jasotzen beste gai batengandik. Baino, informazioa “/cmd_vel” gaiean argitalatzen du beste batzuen artean, robotaren mugimendua ahalbidetuz. Beraz, “/cmd_vel” gaiak eskaintzen duen informazioa erreparatuz:

```
ros2 topic info /cmd_vel
```

Erantzuna:

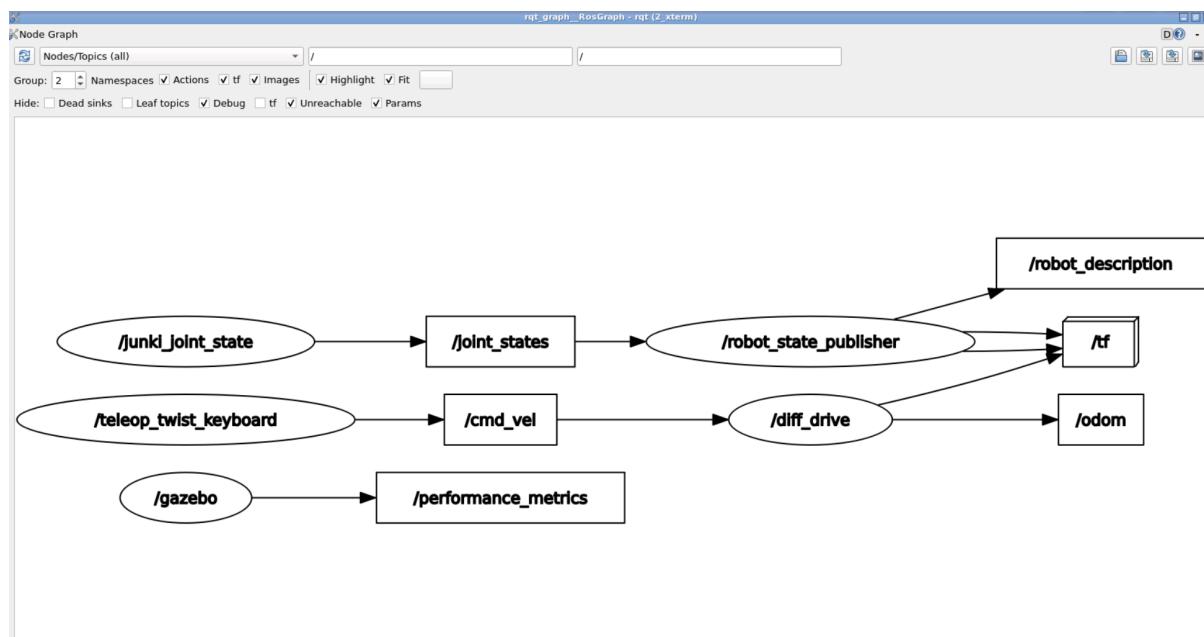
```
Type: geometry_msgs/msg/Twist
Publisher count: 1
Subscription count: 1
```

Komentatu den bezala, /teleop_twist_keyboard nodoa exekutatu denetik, orain “topic” honek argitalatzale bat du, nodo bera hain zuen ere.

4.10 Rqt_graph

Orain arte aipatu diren nodoen eta haien arteko conexioen irudikapen grafiko bat emateko, “rqt_graph” komandoa erabiliko da. Horretarako terminal berri bat ireki:

```
rqt_graph
```



10. Irudia: nodoen conexioen irudikapena.

10. irudian nodoen konexioa ikus daiteke era grafiko batean. Aurreko atalean aipatu den bezala, “/teleop_twist_keyboard” nodoak datuak bidaltzen ditu “/cmd_vel” gaien, eta honi “/diff_drive” nodoa harpidetuta dago.

5. NABEGAZIOA

Nabigazioa, robot bat puntu batetik bestera autonomiaz mugitzeko aukera ematen duen tresna da. Gida honetan Navigation2 [3] erabiliko da. Nabigaziorako robotak bi gauza jakin behar ditu: lehengoa non dagoen da (*localization*), eta bigarrena nolakoa den mapa (*mapping*).

1. Lokalizazioa (*localization*): Honen helburua robotaren egungo posizioa eta orientazioa denbora errealean zehaztasunez kalkulatzea da. Hori lortzeko, odometria eta sentsoreak (laserrak adibidez) erabiltzen dira, robotaren posizio erlatiboa hautemateko eta ulertzeko. Odometria ibilgailu gurpildunen posizioaren estimazioaren azterketa da nabigazioan zehar.
2. Mapaketa (*mapping*): Robota dagoen inguruneko eredu bat sortzeko prozesua da, ingurunearren egitura zehatz bat irudikatuz, oztopoak, hormak, ateak eta abar. Gida honetan bi erabiliko dira, SLAM edo AMCL.

Nabegazioari buruzko informazio gehiago eta zehatzagoa, Nav2ren orrialde ofizialean aurki daiteke [3].

5.1 Lokalizazioa

5.1.1 LiDAR sentsorea

LiDAR sentsorea distantziak neurtzeko eta bere inguran duenaren irudikapenak sortzeko erabiltzen den gailua da, laser argizko pultsuak erabiliz. Beraz, gailu honen bidez une oro jakingo da non dagoen robota.

Gailu hau Junki_bot-ean gehitzeko, xacro fitxategi berri bat eraikiko da deskribapena

izeneko karpetan. Horretarako, terminal bat ireki:

```
cd junki_ws/src/junki_bot/deskribapena/  
touch lidar.xacro
```

Egin klik [lidar.xacro](#) atalean kode osoa ikusteko.

Kodearen azalpena:

Laburbilduz, lehenengo bisualki sortzen da LIDARRA, bere inertziekin, talkekin, loturekin eta abar. Ondoren, LIDAR puglina integratzen da, bere inguruko oztopoei buruzko eskaneoak egin eta datu horiek "/ scan" topikoan argitara ditzan. Gazeboko plugin-ei buruzko informazio gehiago [esteka](#) honetan.

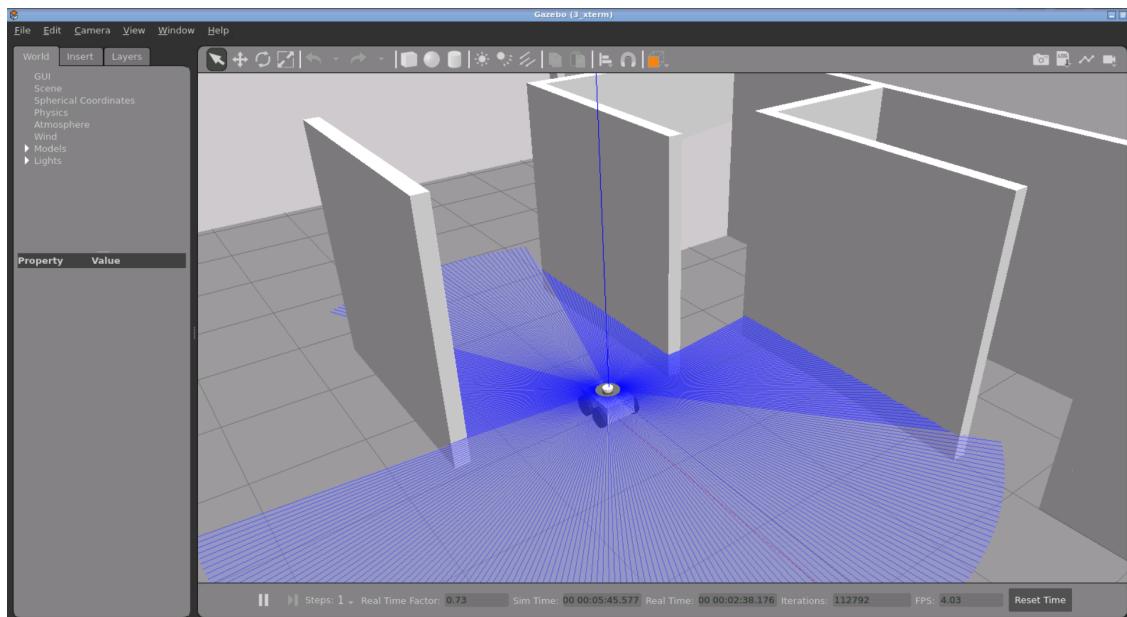
Laser-sentsorea robotean integratzeko, oraindik pausu bat falta da, oraintxe sortutako xacro fitxategia junki_bot.urdf.xacro fitxategian gehitzea:

```
<xacro:include filename="lidar.xacro" />
```

5.1.2 Gazebo

Guzti hau behar bezala konfiguratu dela ziurtatzeko, Gazebon bistaratu da robota. Horretarako terminal batean, konpilatu eta jaurti fitxategia:

```
cd junki_ws/  
colcon build  
source install/setup.bash  
ros2 launch junki_bot gazebo.launch.py
```



11. Irudia: junki_bot-a LIDAR sentsorearekin.

Terminal berri bat ireki:

```
ros2 node list
```

Erantzuna:

```
/diff_drive
/gazebo
/junki_joint_state
/robot_state_publisher
/scan
```

Ikusi daiteke, “/scan” izeneko nodo berri bat agertu dela eta honi buruzko informazioa jasotzeko:

```
ros2 node info /scan
```

Erantzuna:

```
/scan
Subscribers:
/clock: rosgraph_msgs/msg/Clock
/parameter_events: rcl_interfaces/msg/ParameterEvent
Publishers:
/parameter_events: rcl_interfaces/msg/ParameterEvent
/rosout: rcl_interfaces/msg/Log
```

```
/scan: sensor_msgs/msg/LaserScan  
Service Servers:  
/scan/describe_parameters: rcl_interfaces/srv/DescribeParameters  
/scan/get_parameter_types: rcl_interfaces/srv/GetParameterTypes  
/scan/get_parameters: rcl_interfaces/srv/GetParameters  
/scan/list_parameters: rcl_interfaces/srv/ListParameters  
/scan/set_parameters: rcl_interfaces/srv/SetParameters  
/scan/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically  
Service Clients:
```

Action Servers:

Action Clients:

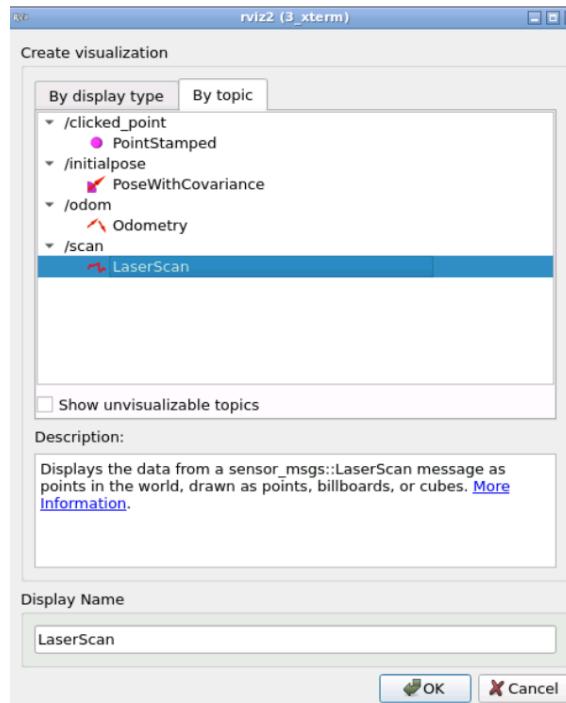
Nodo honen informazioari erreparatuz, ikus daiteke “sensor_msgs/msg/LaserScan” motatako datuak “/scan” izeneko gaiean argitaratzen dituela.

5.1.3 RViz

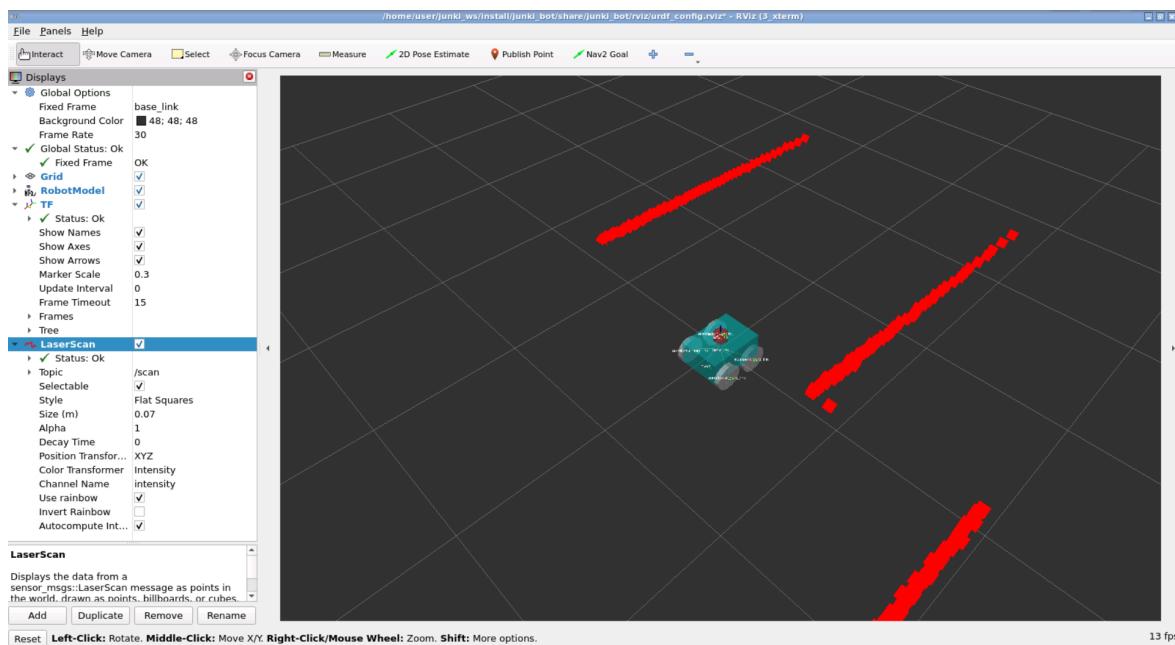
Laserra ondo konfiguratu dela ziurtatzeko, RVizen irekiko da, horretarako beste terminal bat ireki:

```
cd junki_ws/  
source install/setup.bash  
ros2 launch rviz_robota.launch.py
```

Aurreko atalean kodea konpilatu denez eta aldaketarik egin ez denez, bakarrik terminalean paketeak egunerautu dira. Behin, RViz ireki eta robota bertan ageri dela, laserraren funtzionamendua konprobatzeko, egin klik RViz leihoko beheko aldean dagoen “ADD” izeneko botoian. Ondoren, “By topic” atalera joan eta “LaserScan aukera” hautatu 12. irudian ikus daitekeen bezala.



12. Irudia: LaserScan aktibatu RVizen.



13. Irudia: oztopoen detekzioa RVizen, lidarrari esker.

13. irudian ikus daitekeenez, LIDAR sentsorea behar bezala konfiguratu da, eta, horri esker, robota bere ingurunean oztopoak detektatzen ari dela ikus daiteke.

5.2 Mapaketa

5.2.1 Gehitu nabegazio-parametroak

Orain, ROS 2 nabegazio parametroak gehituko dira. Nahiz eta parametro ugari egon, garrantzitsuenak Costmap 2D paketearenak dira. Pakete hau, sare-zelula ugariz osatutako mapa bat da eta Navigation 2 [3] paketeak bi motatakoak erabiltzen ditu inguruneko oztopoen informazioa gordetzeko:

1. Global costmap: kostu-mapa hau epe luzerako planak sortzeko erabiltzen da, adibidez, A puntutik B puntura bitarteko biderik laburrena kalkulatzeko.
2. Local costmap: kostu-mapa hau epe motzeko planak sortzeko erabiltzen da, adibidez, oztopoak saihesteko.

Honi buruzko informazio gehiago, dokumentazio ofizialena aurki daiteke [3].

Terminal bat ireki, eta proiektuaren barruan “params” izeneko karpeta berri bat sortu, eta bere barruan nabegazio-parametroak konfiguratuko duen fitxategia sortu:

```
cd junki_ws/src/junki_bot/  
mkdir params  
cd params  
touch nav2_params.yaml
```

Egin klik [nav2_params.yaml](#) atalean kode osoa ikusteko. Fitxategi hau Junki_bot robotaren nabegazioa planifikatzeko eta kontrolatzeko gaitasunak ematen ditu. Fitxategi honi buruzko informazio gehiago, Nav2ren dokumentazio ofizialean aurki daiteke [3].

Gogoan izan, “params” karpeta berri bat sortu dela eta konpilatu ahal izateko, [CMakeList.txt](#) eguneratu behar dela. Beraz, fitxategi horretan “params” karpeta gehitu.

```
install(  
  DIRECTORY src launch rviz deskribapena worlds params  
  DESTINATION share/${PROJECT_NAME}  
)
```

5.2.2 RViz nabegazioa

Robotarekin nabegatu ahal izateko, RViz fitxategian konfigurazio batzuk egin behar dira. Beraz, terminal bat ireki eta RViz karpetara joan:

```
cd junki_ws/src/junki_bot/rviz/  
touch nav2_default_view.rviz
```

Egin klik [nav2_default_view.rviz](#) atalean kode osoa ikusteko.

Kodearen azalpena:

Laburbilduz fitxategi honetan RViz konfiguratu egin da, nabigazioa ahalbidetuko duten panelen eta bistaratzeen konfigurazioa deskribatuz. Bertan, zer panel erakutsiko diren eta zer bistaratze sartuko diren zehazten dira, eta bistaratze bakoitzaren propietateak eta parametroak ere.

5.2.3 Launch fitxategia

Terminal bat ireki eta “launch” karpetaren barruan fitxategi berri bat sortu:

```
cd junki_ws/src/junki_bot/launch/  
touch rviz_launch.py
```

Fitxategi hau, RViz hasteaz arduratzen da nabigazioa bistaratzea. Egin klik [rviz_launch.py](#) atalean kode osoa ikusteko.

Orain beste fitxategi bat sortuko da “launch” karpetaren barruan, horretarako berriz terminal batean:

```
cd junki_ws/src/junki_bot/launch/  
touch nabegazioa.launch.py
```

RViz aplikazioa abiatuko du, robotaren nabigazioa ahalbidetuko duen konfigurazio guztiarekin. Fitxategi hau funtzionatu dezan, lehenengo gazebo.launch.py fitxategia abiarazi behar da.

Egin klik [nabegazioa.launch.py](#) atalean kode osoa ikusteko.

Kodearen azalpena:

Aurreko “launch” fitxategiekin konparatuz, kasu honetan nabegazioari dagokion konfigurazio batzuk ezarriko dira.

1. “nav2_bringup” direktoriorako bidea bilatu eta biltegiratzen du, aurrerago jaurtiketa-fitxategian erabiltzeko.
2. “DeclareLaunchArgument” klasea erabiliz, SLAM zehaztu da “True” bezala, ingurune ezezagunaren mapa estatiko bat eraikitzea, robota mapa horren barruan mugitzen den bitartean
3. Azkenik, “bringup_launch.py” nabegazio-fitxategia ejekutatzen da, hainbat argudio desberdinak, haien artean SLAM.

Kodea exekutatu aurretik, [package.xml](#) fitxategia eguneratu behar da, nabegazioari buruzko dependentziak gehituz:

```
<build_depend>nav2_common</build_depend>
<build_depend>navigation2</build_depend>
<build_depend>launch_ros</build_depend>

<exec_depend>launch_ros</exec_depend>
<exec_depend>navigation2</exec_depend>
<exec_depend>nav2_common</exec_depend>
<exec_depend>slam_toolbox</exec_depend>
<exec_depend>robot_localization</exec_depend>
```

5.2.4 SLAM exekutatu

Terminal bat ireki eta lan-espazioa konpilatu:

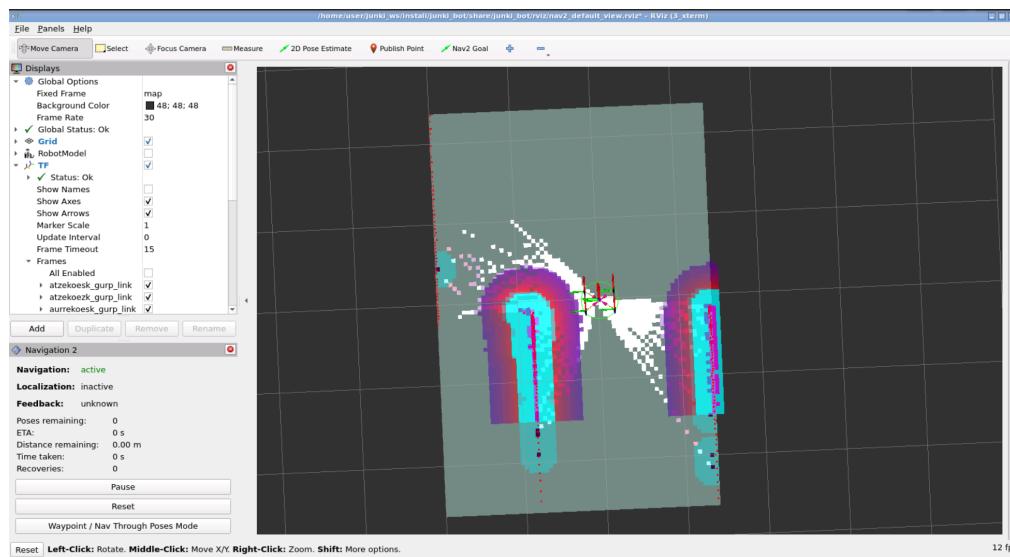
```
cd junki_ws/
colcon build
source install/setup.bash
```

Terminal berri batean, junki_bot-a Gazeboko ingurunean abiarazten duen fitxategia jaurti:

```
ros2 launch junki_bot gazebo.launch.py
```

Fitxategi hau ez da berria; beraz, ikusten denaren adibidea 11. irudian dago. Beste terminal bat ireki:

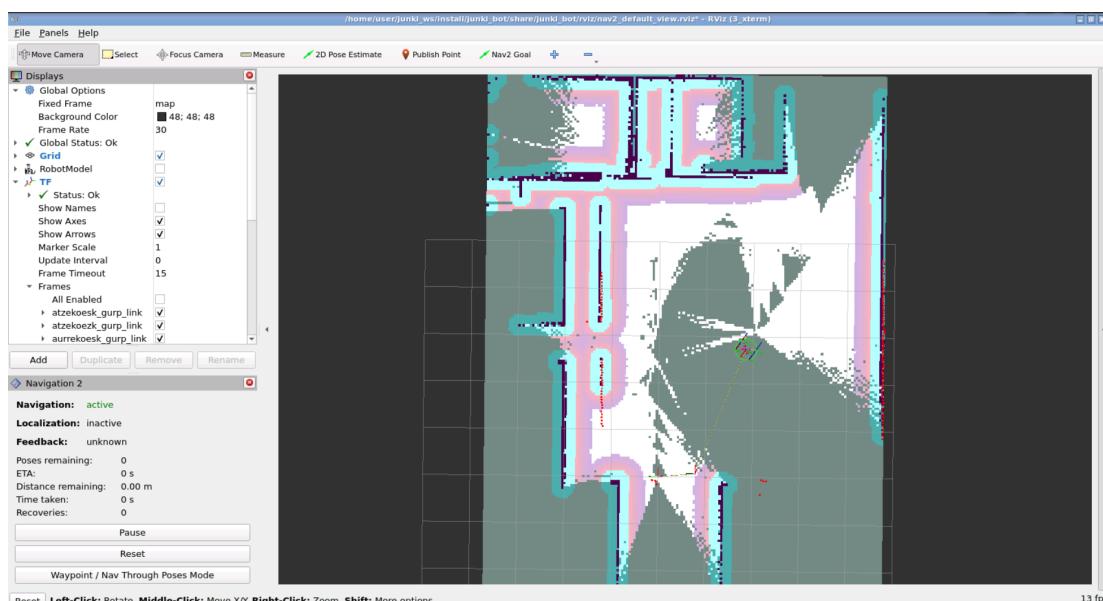
```
source install/setup.bash
ros2 launch junki_bot nabegazioa.launch.py
```



14. Irudia: junki_bot-a SLAMarekin.

14. irudian ikus daitekeen bezala, RViz aplikazioa ireki da Junki_bot-a eta SLAMarekin. Orain mapa estatiko bat sortu nahi da. Horretarako, “teleop_twist_keyboard” exekutagarria abiaraziko da eta teklatua erabiliz robota mugituko da mapa osoaren zehar, mapa guztia kargatzeari. Terminal berri bat ireki:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```



15. Irudia: junki_bot-a SLAMarekin, mapa estatikoa sortzen.

15. irudian ikus daitekeen bezala, robota mapan zehar mugitzen den einean, mapa hau kargatzen ari da. Behin mapa guztiz kargatuta dagoela, mapa estatiko bezala gordetzeko:

Lehenengo maps izeneko karpeta bat sortuko da, horretarako terminal bat ireki:

```
cd junki_ws/src/junki_bot/  
mkdir maps  
cd maps
```

Hurrengo komandoaren bitartez, etxea_mapa.yaml eta etxea_mapa.pgm izeneko fitxategiak sortuko dira:

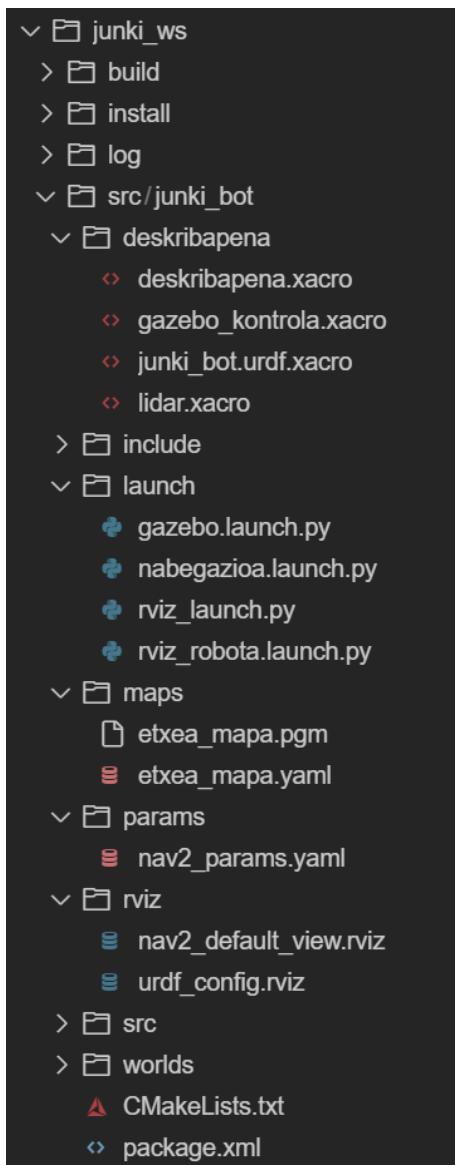
```
ros2 run nav2_map_server map_saver_cli -f etxea_mapa
```

14. Komandoa: mapa estatiko bat sortzeko komandoa.

Kodearen egitura:

```
ros2 run nav2_map_server map_saver_cli -f <maparen izena>
```

Behin bi fitxategi hauek sortu direla, Junki_bot gidako fitxategi guztiak sortuak egongo litzateke. 16. irudian ikus daiteke lan-espazioaren egitura finala.



16. Irudia: Ian-espazioaren egitura.

Gogoan izan, “maps” izeneko karpeta berri bat sortu dela eta konpilatu ahal izateko, [CMakeList.txt](#) eguneratu behar dela. Beraz, fitxategi horretan “params” karpeta gehitu.

```

install(
  DIRECTORY src launch rviz deskrabapena worlds params maps
  DESTINATION share/${PROJECT_NAME} )

```

5.2.5 AMCL

AMCL lokalizazio algoritmo bat da, robotikan erabiltzen dena, robot batek ingurune ezagun batean duen kokapena eta orientazioa kalkulatzeko. Honi buruzko informazio gehiago dokumentazio ofizialean aurki daiteke [3].

Mapa estatiko bat sortu eta gero algoritmo hau implementatu daiteke, horretarako “launch” fitxategia AMCL bezala exekutatuko da, eta hori egin ahal izateko aldaketa txiki batzuk egingo dira [nabegazioa.launch.py](#) fitxategian.

1. Alde batetik, SLAM “False” bezala jarriko da, nabegazioa AMCL bezala exekutatzeko. Kodean “True” bezala dago, aldatu!
2. Bestetik, 5.2.4 atalean sortutako “etxea_mapa.yaml” mapa estatikoa kargatuko da jaurtiketa fitxategitik. Aldaketa hau kodean implementatuta dago.

Behin hau ikusita, lehen bezala, Junki_bot-a bi jaurtiketa fitxategietatik abiaraziko da. Lehenengo, terminal bat ireki eta lan-espazioa konpilatu:

```
cd junki_ws/  
colcon build  
source install/setup.bash
```

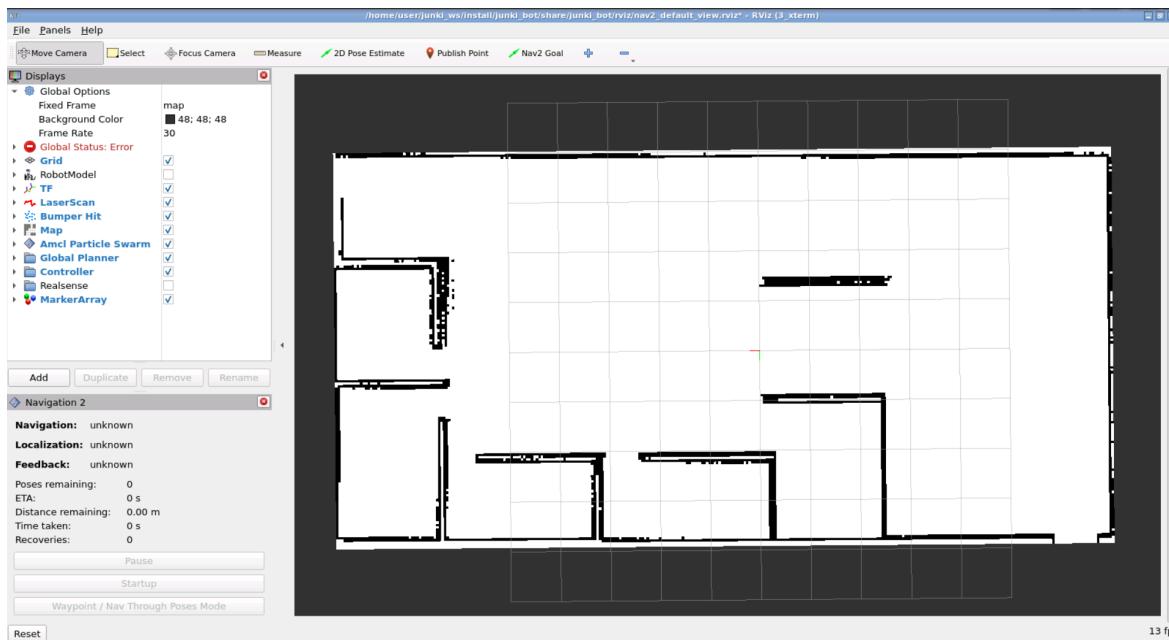
Jaurtiketa fitxategia ezekutatu:

```
ros2 launch junki_bot gazebo.launch.py
```

Fitxategi hau ez da berria; beraz, ikusten denaren adibidea 11. irudian dago. Ondoren, beste terminal bat ireki eta bigarren fitxategia abiarazi:

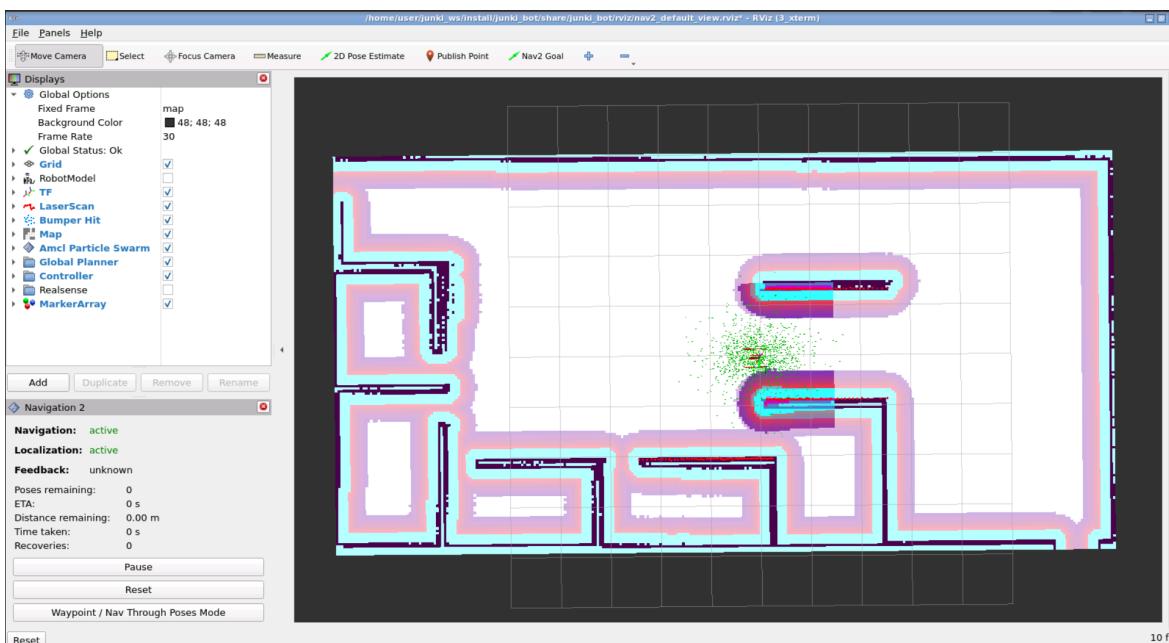
```
source install/setup.bash  
ros2 launch junki_bot nabegazioa.launch.py
```

17. irudian ikus daitekeen bezala, aurretik gorde den mapa estatikoa kargatu da RVizen.



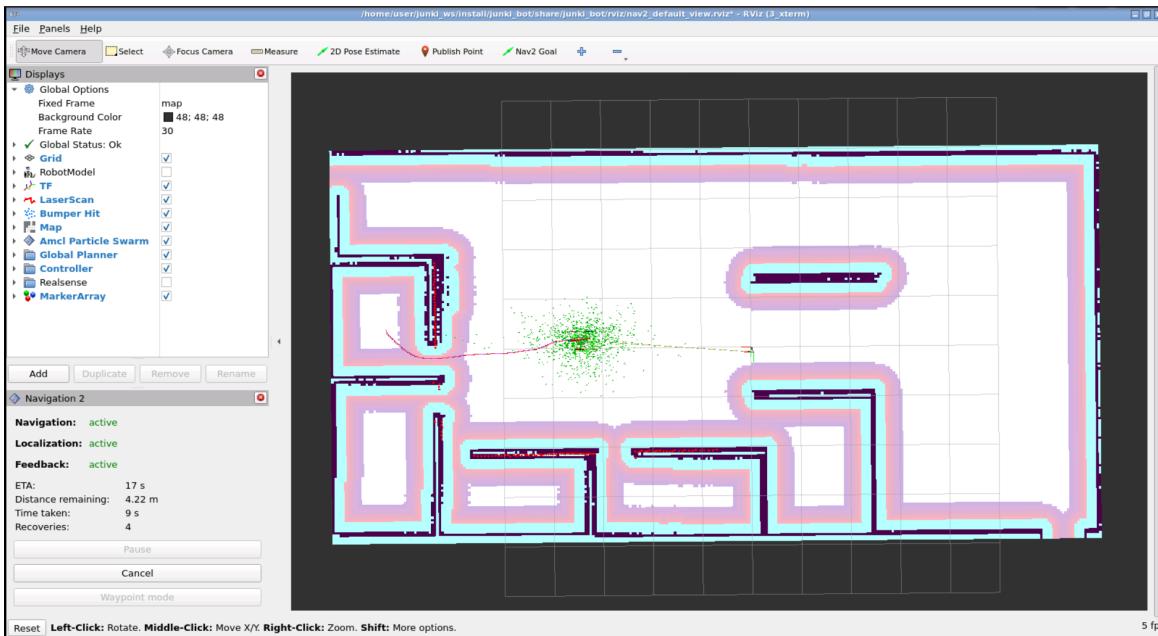
17. Irudia: mapa estatikoa kargatuta RVizen.

Orain nabegatzeko, robotaren posizioa aukeratu behar da. Horretarako, goikaldean dagoen “2D Pose Estimate” izeneko tresna erabiliz, robota mapa horretan aurkitzen den lokalizazioa aukeratu behar.



18. Irudia: AMCL aktibatuta.

Behin hasierako posizioa estimatu dela, robota RVizeko pantailan agertu beharko litzateke eta nabegazioko parametro guztiak aktibatu: “global costmap”, “local costmap” eta abar. Orain, goikaldeko “Nav2 Goal” izeneko tresnaren bitartez, robotari helmuga bat jarriko zaio.



19. Irudia: nabegazioa AMCLrekin.

19. irudian ikusi daitekeke junki_bot-ak helmuga bat duela eta helburu horretara iristeko egin behar duen ibilbidea ageri da.

Sortutako ibilgailuaren koordenatu-markoak ikusi ahal izateko, terminal berri bat ireki:

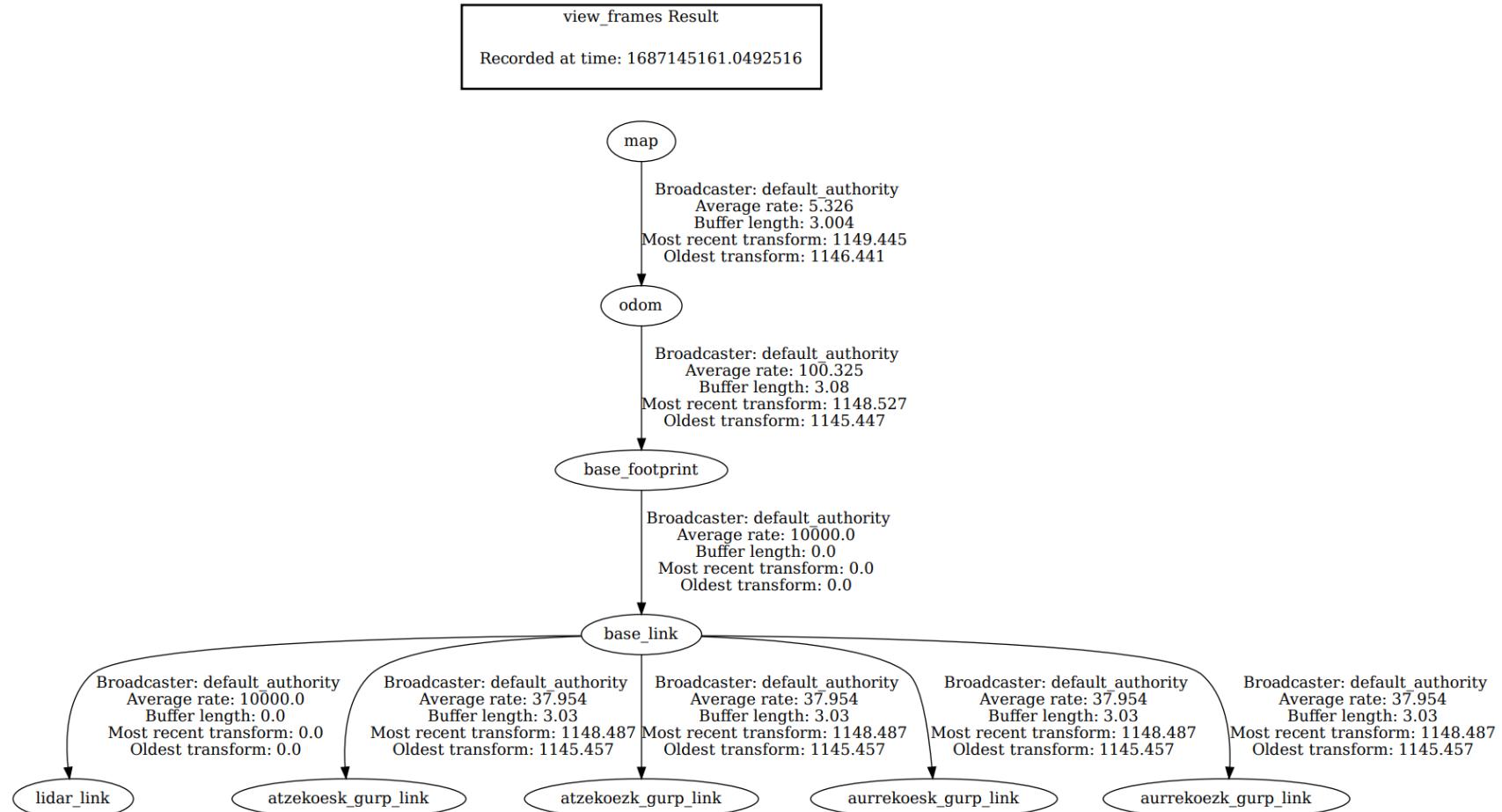
```
ros2 run tf2_tools view_frames
```

Komando hau exekutatu ostean, hurrengo orriko 20. irudian bezala, proiektuko transformatu guztiak zuhaitz bat sortu beharko da. Egitura hurrengoa da:

Map → Odom → Base_footprint → Base_link → ...

Orden hau robotaren kokapena eta orientazioa kalkulatzeko erabilitako transformazioen sekuentzia islatzen du, maparen erreferentzia esparru globalari dagokionez (Map).

Eraldaketa horiei esker, sistemaren osagaietan (sentsoreak, gurpilak, etab.) maparen erreferentzia-esparru orokorrean lan egin dezakete, eta elkarren artean komunikatu, erreferentzia-esparru egokiak erabiliz. Hori funtsezkoa da nabigazio zehatza egiteko eta inguruneko ibilbideak planifikatzeko.



20. Irudia: transformatuen zuhaitza.

6. BIBLIOGRAFIA

[1] The Construct plataforma:

<https://www.theconstructsim.com/>

[2] ROS 2 Humble Hawksbill bertsio ofiziala deskargatzeko:

<https://docs.ros.org/en/humble/Installation.html>

[3] Navigation 2 paketearen dokumentu ofiziala:

<https://docs.ros.org/en/humble/Installation.html>

[4] URDF fitxategiei buruzko dokumentazio ofiziala:

<https://docs.ros.org/en/humble/Tutorials/Intermediate/URDF/URDF-Main.html>

[5] RViz tresnari buruzko dokumentazio ofiziala:

<https://wiki.ros.org/rviz>

[6] Gazebo simulazio-inguruneari buruzko dokumentazio ofiziala:

<https://classic.gazebosim.org/tutorials>

7. ERANSKINA: PROGRAMAZIO-KODEAK

7.1 deskribapena.xacro

```

1 <?xml version="1.0"?>
2 <robot xmlns:xacro="http://www.ros.org/wiki/xacro">
3
4   <!-- Robotaren balio konstanteak -->
5   <xacro:property name="oinarri_zabalera" value="0.31"/>
6   <xacro:property name="oinarri_luzera" value="0.42"/>
7   <xacro:property name="oinarri_altuera" value="0.18"/>
8
9   <xacro:property name="gurpil_erradioa" value="0.10"/>
10  <xacro:property name="gurpil_zabalera" value="0.04"/>
11  <xacro:property name="wheel_ygap" value="0.025"/>
12  <xacro:property name="wheel_zoff" value="0.05"/>
13  <xacro:property name="wheel_xoff" value="0.12"/>
14
15  <!-- Definitu propietate inertzialak -->
16  <xacro:macro name="kutxa_inerzia" params="m w h d">
17    <inertial>
18      <origin xyz="0 0 0" rpy="${pi/2} 0 ${pi/2}" />
19      <mass value="${m}" />
20      <inertia ixx="${(m/12) * (h*h + d*d)}" ixy="0.0" ixz="0.0" iyy="${(m/12) * (w*w + d*d)}" iyz="0.0" izz="${(m/12) * (w*w + h*h)}"/>
21    </inertial>
22  </xacro:macro>
23
24  <xacro:macro name="zilindro_inertzia" params="m r h">
25    <inertial>
26      <origin xyz="0 0 0" rpy="${pi/2} 0 0" />
27      <mass value="${m}" />
28      <inertia ixx="${(m/12) * (3*r*r + h*h)}" ixy = "0" ixz = "0" iyy="${(m/12) * (3*r*r + h*h)}" iyz = "0" izz="${(m/2) * (r*r)}"/>
29    </inertial>
30  </xacro:macro>
31
32  <!-- Robotaren basea -->
33  <link name="base_link">
34    <visual>
35      <geometry>
36        <box size="${oinarri_luzera} ${oinarri_zabalera} ${oinarri_altuera}" />
37      </geometry>
38      <material name="Cyan">
39        <color rgba="0 1.0 1.0 1.0" />
40      </material>
41    </visual>
42    <collision>
43      <geometry>
44        <box size="${oinarri_luzera} ${oinarri_zabalera} ${oinarri_altuera}" />
45      </geometry>
46    </collision>
47
48    <xacro:kutxa_inerzia m="15" w="${oinarri_zabalera}" d="${oinarri_luzera}" h="${oinarri_altuera}" />
49  </link>
50
51
52  <!-- Robotaren Footprinta (itzala) -->
53  <link name="base_footprint" />
54
55

```

```

56 <joint name="base_joint" type="fixed">
57   <parent link="base_footprint"/>
58   <child link="base_link"/>
59   <origin xyz="0.0 0.0 ${-(gurpil_erradioa+wheel_zoff)}" rpy="0 0 0"/>
60 </joint>
61
62 <!-- Atzeko gurpilak -->
63 <xacro:macro name="Atzeko_gurpilak" params="aurrizkia x_norantza y_norantza">
64   <link name="${aurrizkia}_link">
65     <visual>
66       <origin xyz="0 0 0" rpy="${pi/2} 0 0"/>
67       <geometry>
68         <cylinder radius="${gurpil_erradioa}" length="${gurpil_zabalera}"/>
69       </geometry>
70       <material name="txuria">
71         <color rgba="1 1 1 1"/>
72       </material>
73     </visual>
74
75     <collision>
76       <origin xyz="0 0 0" rpy="${pi/2} 0 0"/>
77       <geometry>
78         <cylinder radius="${gurpil_erradioa}" length="${gurpil_zabalera}"/>
79       </geometry>
80     </collision>
81
82   <xacro:zilindro_inertzia m="0.5" r="${gurpil_erradioa}" h="${gurpil_zabalera}"/>
83 </link>
84
85 <joint name="${aurrizkia}_joint" type="continuous">
86   <parent link="base_link"/>
87   <child link="${aurrizkia}_link"/>
88   <origin xyz="${x_norantza*wheel_xoff} ${y_norantza*(oinarri_zabalera/2+wheel_ygap)} ${-wheel_zoff}" rpy="0 0 0"/>
89   <axis xyz="0 1 0"/>
90 </joint>
91 </xacro:macro>
92
93 <xacro:Atzeko_gurpilak aurrizkia="atzekoezk_gurp" x_norantza="-1" y_norantza="1" />
94 <xacro:Atzeko_gurpilak aurrizkia="atzekoesk_gurp" x_norantza="-1" y_norantza="-1" />
95
96 <!-- Aurreko gurpilak -->
97 <xacro:macro name="Aurreko_gurpila" params="aurrizkia2 x_norantza2 y_norantza2">
98   <link name="${aurrizkia2}_link">
99     <visual>
100       <origin xyz="0 0 0" rpy="${pi/2} 0 0"/>
101       <geometry>
102         <cylinder radius="${gurpil_erradioa}" length="${gurpil_zabalera}"/>
103       </geometry>
104       <material name="txuria">
105         <color rgba="1 1 1 1"/>
106       </material>
107     </visual>
108
109     <collision>
110       <origin xyz="0 0 0" rpy="${pi/2} 0 0"/>
111       <geometry>
112         <cylinder radius="${gurpil_erradioa}" length="${gurpil_zabalera}"/>
113       </geometry>
114     </collision>
115
116   <xacro:zilindro_inertzia m="0.5" r="${gurpil_erradioa}" h="${gurpil_zabalera}"/>
117 </link>

```

```

118 <joint name="${aurrizkia2}_joint" type="continuous">
119   <parent link="base_link"/>
120   <child link="${aurrizkia2}_link"/>
121   <origin xyz="${x_norantza2}*wheel_xoff ${y_norantza2}*(oinarri_zabalera/2+wheel_ygap) ${-wheel_zoff}" rpy="0 0 0"/>
122   <axis xyz="0 1 0"/>
123 </joint>
124 </xacro:macro>
125
126 <xacro:Aurreko_gurpila aurrizkia2="urrekoezk_gurp" x_norantza2="1" y_norantza2="1" />
127 <xacro:Aurreko_gurpila aurrizkia2="urrekoesk_gurp" x_norantza2="1" y_norantza2="-1" />
128
129 </robot>

```

1. Programazio-kodea: deskribapena.xacro

7.2 gazebo_kontrol.xacro

```

1 <?xml version="1.0"?>
2 <robot xmlns:xacro="http://www.ros.org/wiki/xacro">
3
4   <gazebo reference="urrekoezk_gurp_link">
5     <material>Gazebo/DarkGrey</material>
6     <mu1>0.9</mu1>
7     <mu2>0.9</mu2>
8   </gazebo>
9
10  <gazebo reference="urrekoesk_gurp_link">
11    <material>Gazebo/DarkGrey</material>
12    <mu1>0.9</mu1>
13    <mu2>0.9</mu2>
14  </gazebo>
15
16  <gazebo reference="atzekoezk_gurp_link">
17    <material>Gazebo/DarkGrey</material>
18    <mu1>0.9</mu1>
19    <mu2>0.9</mu2>
20  </gazebo>
21
22  <gazebo reference="atzekoesk_gurp_link">
23    <material>Gazebo/DarkGrey</material>
24    <mu1>0.9</mu1>
25    <mu2>0.9</mu2>
26  </gazebo>
27
28  <gazebo>
29    <plugin name='diff_drive' filename='libgazebo_ros_diff_drive.so'>
30      <ros>
31        <remapping>/tf:=tf</remapping>
32      </ros>
33
34      <update_rate>100.0</update_rate>
35      <num_wheel_pairs>2</num_wheel_pairs>
36
37      <left_joint>urrekoezk_gurp_joint</left_joint>
38      <left_joint>atzekoezk_gurp_joint</left_joint>
39      <wheel_separation>0.4</wheel_separation>
40      <wheel_diameter>0.2</wheel_diameter>
41
42      <right_joint>atzekoesk_gurp_joint</right_joint>
43      <right_joint>urrekoesk_gurp_joint</right_joint>

```

```

44 <wheel_separation>0.4</wheel_separation>
45 <wheel_diameter>0.2</wheel_diameter>
46 <!-- limiteak -->
47 <max_wheel_torque>20</max_wheel_torque>
48 <max_wheel_acceleration>1.0</max_wheel_acceleration>
49
50 <command_topic>cmd_vel</command_topic>
51
52 <!-- irteera -->
53 <publish_odom>true</publish_odom>
54 <publish_odom_tf>true</publish_odom_tf>
55 <publish_wheel_tf>false</publish_wheel_tf>
56 <odometry_topic>odom</odometry_topic>
57 <odometry_frame>odom</odometry_frame>
58 <robot_base_frame>base_footprint</robot_base_frame>
59 <!-- Odometry source, 0 for ENCODER, 1 for WORLD, defaults to WORLD -->
60 <odometry_source>1</odometry_source>
61 </plugin>
62
63 <plugin name="junki_joint_state" filename="libgazebo_ros_joint_state_publisher.so">
64 <ros>
65   <!-- <namespace>/tb3</namespace> -->
66   <remapping>~/out:=joint_states</remapping>
67 </ros>
68 <update_rate>100.0</update_rate>
69 <joint_name>aurrekoezk_gurp_joint</joint_name>
70 <joint_name>atzekoezk_gurp_joint</joint_name>
71 <joint_name>atzekoesk_gurp_joint</joint_name>
72 <joint_name>aurrekoesk_gurp_joint</joint_name>
73 </plugin>
74 </gazebo>
75 </robot>

```

2. Programazio-kodea: gazebo_kontrol.xacro

7.3 junki_bot.urdf.xacro

```

1 <?xml version="1.0"?>
2 <robot xmlns:xacro="http://www.ros.org/wiki/xacro" name="junki_bot">
3
4   <xacro:include filename="deskribapena.xacro" />
5   <xacro:include filename="gazebo_kontrola.xacro" />
6
7 </robot>

```

3. Programazio-kodea: junki_bot.urdf.xacro

7.4 rviz_robota.launch.py

```

1 import os
2 from launch import LaunchDescription
3 from launch.actions import DeclareLaunchArgument
4 from launch.conditions import IfCondition, UnlessCondition
5 from launch.substitutions import Command, LaunchConfiguration
6 from launch_ros.actions import Node
7 from launch_ros.substitutions import FindPackageShare
8
9 def generate_launch_description():
10
11   # Set the path to different files and folders.

```

```

12 pkg_share = FindPackageShare(package='junki_bot').find('junki_bot')
13 default_model_path = os.path.join(pkg_share, 'deskribapena/junki_bot.urdf.xacro')
14 default_rviz_config_path = os.path.join(pkg_share, 'rviz/rviz_config.rviz')
15
16 # Launch configuration variables specific to simulation
17 gui = LaunchConfiguration('gui')
18 model = LaunchConfiguration('model')
19 rviz_config_file = LaunchConfiguration('rviz_config_file')
20 use_robot_state_pub = LaunchConfiguration('use_robot_state_pub')
21 use_rviz = LaunchConfiguration('use_rviz')
22 use_sim_time = LaunchConfiguration('use_sim_time')
23
24 # Declare the launch arguments
25 declare_model_path_cmd = DeclareLaunchArgument(
26   name='model',
27   default_value=default_model_path,
28   description='Absolute path to robot urdf file')
29
30 declare_rviz_config_file_cmd = DeclareLaunchArgument(
31   name='rviz_config_file',
32   default_value=default_rviz_config_path,
33   description='Full path to the RVIZ config file to use')
34
35 declare_use_joint_state_publisher_cmd = DeclareLaunchArgument(
36   name='gui',
37   default_value='True',
38   description='Flag to enable joint_state_publisher_gui')
39
40 declare_use_robot_state_pub_cmd = DeclareLaunchArgument(
41   name='use_robot_state_pub',
42   default_value='True',
43   description='Whether to start the robot state publisher')
44
45
46 declare_use_rviz_cmd = DeclareLaunchArgument(
47   name='use_rviz',
48   default_value='True',
49   description='Whether to start RVIZ')
50
51 declare_use_sim_time_cmd = DeclareLaunchArgument(
52   name='use_sim_time',
53   default_value='True',
54   description='Use simulation (Gazebo) clock if true')
55
56 # Specify the actions
57
58 # Publish the joint state values for the non-fixed joints in the URDF file.
59 start_joint_state_publisher_cmd = Node(
60   condition=UnlessCondition(gui),
61   package='joint_state_publisher',
62   executable='joint_state_publisher',
63   name='joint_state_publisher')
64
65 # A GUI to manipulate the joint state values
66 start_joint_state_publisher_gui_node = Node(
67   condition=IfCondition(gui),
68   package='joint_state_publisher_gui',
69   executable='joint_state_publisher_gui',
70   name='joint_state_publisher_gui')
71
72 # Subscribe to the joint states of the robot, and publish the 3D pose of each link.
73 start_robot_state_publisher_cmd = Node(

```

```

74 condition=IfCondition(use_robot_state_pub),
75 package='robot_state_publisher',
76 executable='robot_state_publisher',
77 parameters=[{"use_sim_time": use_sim_time,
78 'robot_description': Command(['xacro ', model])}],
79 arguments=[default_model_path])
80
81 # Launch RViz
82 start_rviz_cmd = Node(
83   condition=IfCondition(use_rviz),
84   package='rviz2',
85   executable='rviz2',
86   name='rviz2',
87   output='screen',
88   arguments=['-d', rviz_config_file])
89
90 # Create the launch description and populate
91 ld = LaunchDescription()
92
93 # Declare the launch options
94 ld.add_action(declare_model_path_cmd)
95 ld.add_action(declare_rviz_config_file_cmd)
96 ld.add_action(declare_use_joint_state_publisher_cmd)
97 ld.add_action(declare_use_robot_state_pub_cmd)
98 ld.add_action(declare_use_rviz_cmd)
99 ld.add_action(declare_use_sim_time_cmd)
100
101 # Add any actions
102 ld.add_action(start_joint_state_publisher_cmd)
103 ld.add_action(start_joint_state_publisher_gui_node)
104 ld.add_action(start_robot_state_publisher_cmd)
105 ld.add_action(start_rviz_cmd)
106
107 return ld

```

4. Programazio-kodea: rviz_robota.launch.py

7.5 urdf_config.rviz

```

1 Panels:
2   - Class: rviz_common/Displays
3     Help Height: 78
4     Name: Displays
5     Property Tree Widget:
6       Expanded:
7         - /Global Options1
8         - /Status1
9         - /RobotModel1/Links1
10        - /TF1
11     Splitter Ratio: 0.5
12     Tree Height: 557
13 Visualization Manager:
14   Class: ""
15   Displays:
16     - Alpha: 0.5
17     Cell Size: 1
18     Class: rviz_default_plugins/Grid
19     Color: 160; 160; 164
20     Enabled: true
21     Name: Grid
22     - Alpha: 0.6

```

```

23   Class: rviz_default_plugins/RobotModel
24     Description Topic:
25       Depth: 5
26       Durability Policy: Volatile
27       History Policy: Keep Last
28       Reliability Policy: Reliable
29       Value: /robot_description
30     Enabled: true
31     Name: RobotModel
32     Visual Enabled: true
33     - Class: rviz_default_plugins/TF
34       Enabled: true
35       Name: TF
36       Marker Scale: 0.3
37       Show Arrows: true
38       Show Axes: true
39       Show Names: true
40     Enabled: true
41     Global Options:
42       Background Color: 48; 48; 48
43       Fixed Frame: base_link
44       Frame Rate: 30
45     Name: root
46     Tools:
47     - Class: rviz_default_plugins/Interact
48       Hide Inactive Objects: true
49     - Class: rviz_default_plugins/MoveCamera
50     - Class: rviz_default_plugins/Select
51     - Class: rviz_default_plugins/FocusCamera
52     - Class: rviz_default_plugins/Measure
53     Line color: 128; 128; 0
54     - Class: rviz_default_plugins/SetInitialPose
55     Topic:
56       Depth: 5
57       Durability Policy: Volatile
58       History Policy: Keep Last
59       Reliability Policy: Reliable
60       Value: /initialpose
61     - Class: rviz_default_plugins/PublishPoint
62       Single click: true
63     Topic:
64       Depth: 5
65       Durability Policy: Volatile
66       History Policy: Keep Last
67       Reliability Policy: Reliable
68       Value: /clicked_point
69     - Class: nav2_rviz_plugins/GoalTool
70   Transformation:
71     Current:
72       Class: rviz_default_plugins/TF
73     Value: true
74   Views:
75     Current:
76       Class: rviz_default_plugins/Orbit
77       Name: Current View
78     Target Frame: <Fixed Frame>
79     Value: Orbit (rviz)
80   Saved: ~
  
```

5. Programazio-kodea: urdf_config.rviz

7.6 package.xml

```

1 <?xml version="1.0"?>
2 <?xml-model href="http://download.ros.org/schema/package_format3.xsd"
3 schematypens="http://www.w3.org/2001/XMLSchema"?>
4 <package format="3">
5   <name>junki_bot</name>
6   <version>0.0.0</version>
7   <description>TODO: Package description</description>
8   <maintainer email="user@todo.todo">user</maintainer>
9   <license>TODO: License declaration</license>
10
11  <buildtool_depend>ament_cmake</buildtool_depend>
12
13  <exec_depend>joint_state_publisher</exec_depend>
14  <exec_depend>robot_state_publisher</exec_depend>
15  <exec_depend>rviz2</exec_depend>
16  <exec_depend>xacro</exec_depend>
17  <test_depend>ament_lint_auto</test_depend>
18  <test_depend>ament_lint_common</test_depend>
19
20  <export>
21    <build_type>ament_cmake</build_type>
22  </export>
23 </package>

```

6. Programazio-kodea: package.xml

7.7 CMakeLists.txt

```

1 cmake_minimum_required(VERSION 3.8)
2 project(junki_bot)
3
4 if(CMAKE_COMPILER_IS_GNUCXX OR CMAKE_CXX_COMPILER_ID MATCHES "Clang")
5   add_compile_options(-Wall -Wextra -Wpedantic)
6 endif()
7
8 # find dependencies
9 find_package(ament_cmake REQUIRED)
10 # uncomment the following section in order to fill in
11 # further dependencies manually.
12 # find_package(<dependency> REQUIRED)
13
14 install(
15   DIRECTORY src launch rviz deskribapena
16   DESTINATION share/${PROJECT_NAME}
17 )
18
19 if(BUILD_TESTING)
20   find_package(ament_lint_auto REQUIRED)
21   # the following line skips the linter which checks for copyrights
22   # comment the line when a copyright and license is added to all source files
23   set(ament_cmake_copyright_FOUND TRUE)
24   # the following line skips cpplint (only works in a git repo)
25   # comment the line when this package is in a git repo and when
26   # a copyright and license is added to all source files
27   set(ament_cmake_cpplint_FOUND TRUE)
28   ament_lint_auto_find_test_dependencies()
29 endif()
30 ament_package()

```

7. Programazio-kodea: CMakeLists.txt

7.8 etxea.world

```

1 <sdf version='1.6'>
2   <world name='default'>
3     <light name='sun' type='directional'>
4       <cast_shadows>1</cast_shadows>
5       <pose frame="">0 0 10 0 -0 0</pose>
6       <diffuse>0.8 0.8 0.8 1</diffuse>
7       <specular>0.2 0.2 0.2 1</specular>
8       <attenuation>
9         <range>1000</range>
10        <constant>0.9</constant>
11        <linear>0.01</linear>
12        <quadratic>0.001</quadratic>
13     </attenuation>
14     <direction>-0.5 0.1 -0.9</direction>
15   </light>
16   <model name='ground_plane'>
17     <static>1</static>
18     <link name='link'>
19       <collision name='collision'>
20         <geometry>
21           <plane>
22             <normal>0 0 1</normal>
23             <size>100 100</size>
24           </plane>
25         </geometry>
26         <surface>
27           <contact>
28             <collide_bitmask>65535</collide_bitmask>
29             <ode/>
30           </contact>
31           <friction>
32             <ode>
33               <mu>100</mu>
34               <mu2>50</mu2>
35             </ode>
36             <torsional>
37               <ode/>
38             </torsional>
39           </friction>
40           <bounce/>
41         </surface>
42         <max_contacts>10</max_contacts>
43       </collision>
44       <visual name='visual'>
45         <cast_shadows>0</cast_shadows>
46         <geometry>
47           <plane>
48             <normal>0 0 1</normal>
49             <size>100 100</size>
50           </plane>
51         </geometry>
52         <material>
53           <script>
54             <uri>file:///media/materials/scripts/gazebo.material</uri>
55             <name>Gazebo/Grey</name>
56           </script>
57         </material>
58       </visual>
59       <self_collide>0</self_collide>
60       <enable_wind>0</enable_wind>

```

```

61      <kinematic>0</kinematic>
62      </link>
63    </model>
64    <gravity>0 0 -9.8</gravity>
65    <magnetic_field>6e-06 2.3e-05 -4.2e-05</magnetic_field>
66    <atmosphere type='adiabatic' />
67    <physics name='default_physics' default='0' type='ode'>
68      <max_step_size>0.001</max_step_size>
69      <real_time_factor>1</real_time_factor>
70      <real_time_update_rate>1000</real_time_update_rate>
71    </physics>
72    <scene>
73      <ambient>0.4 0.4 0.4 1</ambient>
74      <background>0.7 0.7 0.7 1</background>
75      <shadows>1</shadows>
76    </scene>
77    <audio>
78      <device>default</device>
79    </audio>
80    <wind/>
81    <spherical_coordinates>
82      <surface_model>EARTH_WGS84</surface_model>
83      <latitude_deg>0</latitude_deg>
84      <longitude_deg>0</longitude_deg>
85      <elevation>0</elevation>
86      <heading_deg>0</heading_deg>
87    </spherical_coordinates>
88    <model name='sudhir.building'>
89      <pose frame="">-1.27854 7.44696 0 0 -0 0</pose>
90      <link name='Wall_0'>
91        <collision name='Wall_0_Collision'>
92          <geometry>
93            <box>
94              <size>4 0.15 2.5</size>
95            </box>
96          </geometry>
97          <pose frame="">0 0 1.25 0 -0 0</pose>
98          <max_contacts>10</max_contacts>
99          <surface>
100            <contact>
101              <ode/>
102            </contact>
103            <bounce/>
104            <friction>
105              <torsional>
106                <ode/>
107              </torsional>
108                <ode/>
109            </friction>
110          </surface>
111        </collision>
112        <visual name='Wall_0_Visual'>
113          <pose frame="">0 0 1.25 0 -0 0</pose>
114          <geometry>
115            <box>
116              <size>4 0.15 2.5</size>
117            </box>
118          </geometry>
119          <meta>
120            <layer>0</layer>
121          </meta>
122        </visual>

```

```

123 <pose frame="">-7.92943 1.85 0 0 -0 -1.5708</pose>
124 <self_collide>0</self_collide>
125 <enable_wind>0</enable_wind>
126 <kinematic>0</kinematic>
127 </link>
128
129 <link name="Wall_11">
130   <collision name="Wall_11_Collision">
131     <geometry>
132       <box>
133         <size>2.5 0.15 2.5</size>
134       </box>
135     </geometry>
136     <pose frame="">0 0 1.25 0 -0 0</pose>
137     <max_contacts>10</max_contacts>
138   <surface>
139     <contact>
140       <ode/>
141     </contact>
142     <bounce/>
143     <friction>
144       <torsional>
145         <ode/>
146       </torsional>
147       <ode/>
148     </friction>
149   </surface>
150 </collision>
151 <visual name="Wall_11_Visual">
152   <pose frame="">0 0 1.25 0 -0 0</pose>
153   <geometry>
154     <box>
155       <size>2.5 0.15 2.5</size>
156     </box>
157   </geometry>
158   <meta>
159     <layer>0</layer>
160   </meta>
161 </visual>
162 <pose frame="">-2.06679 -1.31994 0 0 -0 3.14159</pose>
163 <self_collide>0</self_collide>
164 <enable_wind>0</enable_wind>
165 <kinematic>0</kinematic>
166 </link>
167 <link name="Wall_13">
168   <collision name="Wall_13_Collision">
169     <geometry>
170       <box>
171         <size>1.75 0.15 2.5</size>
172       </box>
173     </geometry>
174     <pose frame="">0 0 1.25 0 -0 0</pose>
175     <max_contacts>10</max_contacts>
176   <surface>
177     <contact>
178       <ode/>
179     </contact>
180     <bounce/>
181     <friction>
182       <torsional>
183         <ode/>
184       </torsional>
  
```

```

185      <ode/>
186      </friction>
187      </surface>
188      </collision>
189      <visual name="Wall_13_Visual">
190        <pose frame="">0 0 1.25 0 -0 0</pose>
191        <geometry>
192          <box>
193            <size>1.75 0.15 2.5</size>
194          </box>
195        </geometry>
196
197        <meta>
198          <layer>0</layer>
199        </meta>
200      </visual>
201      <pose frame="">-0.908613 2.9996 0 0 -0 -1.5708</pose>
202      <self_collide>0</self_collide>
203      <enable_wind>0</enable_wind>
204      <kinematic>0</kinematic>
205      </link>
206      <link name="Wall_14">
207        <collision name="Wall_14_Collision">
208          <geometry>
209            <box>
210              <size>2.75 0.15 2.5</size>
211            </box>
212          </geometry>
213          <pose frame="">0 0 1.25 0 -0 0</pose>
214          <max_contacts>10</max_contacts>
215          <surface>
216            <contact>
217              <ode/>
218            </contact>
219            <bounce/>
220            <friction>
221              <torsional>
222                <ode/>
223              </torsional>
224            <ode/>
225            </friction>
226          </surface>
227        </collision>
228        <visual name="Wall_14_Visual">
229          <pose frame="">0 0 1.25 0 -0 0</pose>
230          <geometry>
231            <box>
232              <size>2.75 0.15 2.5</size>
233            </box>
234          </geometry>
235
236          <meta>
237            <layer>0</layer>
238          </meta>
239        </visual>
240        <pose frame="">0.391387 2.19961 0 0 -0 0</pose>
241        <self_collide>0</self_collide>
242        <enable_wind>0</enable_wind>
243        <kinematic>0</kinematic>
244        </link>
245        <link name="Wall_17">
246          <collision name="Wall_17_Collision">

```

```

247 <geometry>
248   <box>
249     <size>1.75 0.15 2.5</size>
250   </box>
251 </geometry>
252 <pose frame="">0 0 1.25 0 -0 0</pose>
253 <max_contacts>10</max_contacts>
254 <surface>
255   <contact>
256     <ode/>
257   </contact>
258   <bounce/>
259   <friction>
260     <torsional>
261       <ode/>
262     </torsional>
263       <ode/>
264     </friction>
265   </surface>
266 </collision>
267 <visual name="Wall_17_Visual">
268   <pose frame="">0 0 1.25 0 -0 0</pose>
269   <geometry>
270     <box>
271       <size>1.75 0.15 2.5</size>
272     </box>
273   </geometry>
274   <meta>
275     <layer>0</layer>
276   </meta>
277 </visual>
278 <pose frame="">2.63138 3.01972 0 0 -0 -1.5708</pose>
279 <self_collide>0</self_collide>
280 <enable_wind>0</enable_wind>
281 <kinematic>0</kinematic>
282 </link>
283 <link name="Wall_18">
284   <collision name="Wall_18_Collision">
285     <geometry>
286       <box>
287         <size>2.5 0.15 2.5</size>
288       </box>
289     </geometry>
290   <pose frame="">0 0 1.25 0 -0 0</pose>
291   <max_contacts>10</max_contacts>
292   <surface>
293     <contact>
294       <ode/>
295     </contact>
296     <bounce/>
297     <friction>
298       <torsional>
299         <ode/>
300       </torsional>
301       <ode/>
302     </friction>
303   </surface>
304 </collision>
305 <visual name="Wall_18_Visual">
306   <pose frame="">0 0 1.25 0 -0 0</pose>
307   <geometry>
308     <box>
  
```

```

309      <size>2.5 0.15 2.5</size>
310      </box>
311      </geometry>
312      <meta>
313        <layer>0</layer>
314      </meta>
315      </visual>
316      <pose frame="">3.80638 2.21972 0 0 -0 0</pose>
317      <self_collide>0</self_collide>
318      <enable_wind>0</enable_wind>
319      <kinematic>0</kinematic>
320      </link>
321      <link name='Wall_20'>
322        <collision name='Wall_20_Collision'>
323          <geometry>
324            <box>
325              <size>2.5 0.15 2.5</size>
326            </box>
327          </geometry>
328          <pose frame="">0 0 1.25 0 -0 0</pose>
329          <max_contacts>10</max_contacts>
330          <surface>
331            <contact>
332              <ode/>
333            </contact>
334            <bounce/>
335            <friction>
336              <torsional>
337                <ode/>
338              </torsional>
339                <ode/>
340            </friction>
341          </surface>
342        </collision>
343        <visual name='Wall_20_Visual'>
344          <pose frame="">0 0 1.25 0 -0 0</pose>
345          <geometry>
346            <box>
347              <size>2.5 0.15 2.5</size>
348            </box>
349          </geometry>
350          <meta>
351            <layer>0</layer>
352          </meta>
353        </visual>
354        <pose frame="">5.74898 2.68495 0 0 -0 -1.5708</pose>
355        <self_collide>0</self_collide>
356        <enable_wind>0</enable_wind>
357        <kinematic>0</kinematic>
358        </link>
359        <link name='Wall_22'>
360          <collision name='Wall_22_Collision'>
361            <geometry>
362              <box>
363                <size>2.25 0.15 2.5</size>
364              </box>
365            </geometry>
366            <pose frame="">0 0 1.25 0 -0 0</pose>
367            <max_contacts>10</max_contacts>
368            <surface>
369              <contact>
370                <ode/>

```

```

371   </contact>
372   <bounce/>
373   <friction>
374     <torsional>
375       <ode/>
376     </torsional>
377     <ode/>
378   </friction>
379   </surface>
380 </collision>
381 <visual name='Wall_22_Visual'>
382   <pose frame='>0 0 1.25 0 -0 0</pose>
383   <geometry>
384     <box>
385       <size>2.25 0.15 2.5</size>
386     </box>
387   </geometry>
388   <meta>
389     <layer>0</layer>
390   </meta>
391 </visual>
392 <pose frame='>6.81909 0.722229 0 0 -0 0</pose>
393 <self_collide>0</self_collide>
394 <enable_wind>0</enable_wind>
395 <kinematic>0</kinematic>
396 </link>
397 <link name='Wall_24'>
398   <collision name='Wall_24_Collision'>
399     <geometry>
400       <box>
401         <size>1.75 0.15 2.5</size>
402       </box>
403     </geometry>
404   <pose frame='>0 0 1.25 0 -0 0</pose>
405   <max_contacts>10</max_contacts>
406   <surface>
407     <contact>
408       <ode/>
409     </contact>
410     <bounce/>
411     <friction>
412       <torsional>
413         <ode/>
414       </torsional>
415         <ode/>
416       </friction>
417     </surface>
418   </collision>
419   <visual name='Wall_24_Visual'>
420     <pose frame='>0 0 1.25 0 -0 0</pose>
421     <geometry>
422       <box>
423         <size>1.75 0.15 2.5</size>
424       </box>
425     </geometry>
426     <meta>
427       <layer>0</layer>
428     </meta>
429   </visual>
430   <pose frame='>5.82943 -0.902428 0 0 -0 -1.5708</pose>
431   <self_collide>0</self_collide>
432   <enable_wind>0</enable_wind>

```

```

433   <kinematic>0</kinematic>
434   </link>
435   <link name='Wall_25'>
436     <collision name='Wall_25_Collision'>
437       <geometry>
438         <box>
439           <size>2.25 0.15 2.5</size>
440         </box>
441       </geometry>
442       <pose frame="">0 0 1.25 0 -0 0</pose>
443       <max_contacts>10</max_contacts>
444       <surface>
445         <contact>
446           <ode/>
447         </contact>
448         <bounce/>
449         <friction>
450           <torsional>
451             <ode/>
452           </torsional>
453             <ode/>
454           </friction>
455         </surface>
456       </collision>
457       <visual name='Wall_25_Visual'>
458         <pose frame="">0 0 1.25 0 -0 0</pose>
459         <geometry>
460           <box>
461             <size>2.25 0.15 2.5</size>
462           </box>
463         </geometry>
464         <meta>
465           <layer>0</layer>
466         </meta>
467       </visual>
468       <pose frame="">6.87943 -1.70243 0 0 -0 0</pose>
469       <self_collide>0</self_collide>
470       <enable_wind>0</enable_wind>
471       <kinematic>0</kinematic>
472     </link>
473     <link name='Wall_3'>
474       <collision name='Wall_3_Collision'>
475         <geometry>
476           <box>
477             <size>4 0.15 2.5</size>
478           </box>
479         </geometry>
480         <pose frame="">0 0 1.25 0 -0 0</pose>
481         <max_contacts>10</max_contacts>
482         <surface>
483           <contact>
484             <ode/>
485           </contact>
486           <bounce/>
487           <friction>
488             <torsional>
489               <ode/>
490             </torsional>
491             <ode/>
492           </friction>
493         </surface>
494       </collision>
  
```

```

495   <visual name="Wall_3_Visual">
496     <pose frame="">0 0 1.25 0 -0 0</pose>
497     <geometry>
498       <box>
499         <size>4 0.15 2.5</size>
500       </box>
501     </geometry>
502     <meta>
503       <layer>0</layer>
504     </meta>
505   </visual>
506   <pose frame="">-7.92943 -2 0 0 -0 -1.5708</pose>
507   <self_collide>0</self_collide>
508   <enable_wind>0</enable_wind>
509   <kinematic>0</kinematic>
510 </link>
511 <link name='Wall_4'>
512   <collision name='Wall_4_Collision'>
513     <geometry>
514       <box>
515         <size>16 0.15 2.5</size>
516       </box>
517     </geometry>
518     <pose frame="">0 0 1.25 0 -0 0</pose>
519     <max_contacts>10</max_contacts>
520     <surface>
521       <contact>
522         <ode/>
523       </contact>
524       <bounce/>
525       <friction>
526         <torsional>
527           <ode/>
528         </torsional>
529           <ode/>
530         </friction>
531       </surface>
532     </collision>
533   <visual name='Wall_4_Visual'>
534     <pose frame="">0 0 1.25 0 -0 0</pose>
535     <geometry>
536       <box>
537         <size>16 0.15 2.5</size>
538       </box>
539     </geometry>
540     <meta>
541       <layer>0</layer>
542     </meta>
543   </visual>
544   <pose frame="">-0.004434 -3.925 0 0 -0 0</pose>
545   <self_collide>0</self_collide>
546   <enable_wind>0</enable_wind>
547   <kinematic>0</kinematic>
548 </link>
549 <link name='Wall_5'>
550   <pose frame="">7.92057 0 0 0 -0 1.5708</pose>
551   <visual name='Wall_5_Visual_0'>
552     <pose frame="">-3.92553 0 1.25 0 -0 0</pose>
553     <geometry>
554       <box>
555         <size>0.148947 0.15 2.5</size>
556       </box>

```

```

557 </geometry>
558 <meta>
559   <layer>0</layer>
560 </meta>
561 </visual>
562 <collision name="Wall_5_Collision_0">
563   <geometry>
564     <box>
565       <size>0.148947 0.15 2.5</size>
566     </box>
567   </geometry>
568   <pose frame="">-3.92553 0 1.25 0 -0 0</pose>
569   <max_contacts>10</max_contacts>
570   <surface>
571     <contact>
572       <ode/>
573     </contact>
574     <bounce/>
575     <friction>
576       <torsional>
577         <ode/>
578       </torsional>
579         <ode/>
580     </friction>
581   </surface>
582 </collision>
583 <visual name="Wall_5_Visual_1">
584   <pose frame="">0.524473 0 1.25 0 -0 0</pose>
585   <geometry>
586     <box>
587       <size>6.95105 0.15 2.5</size>
588     </box>
589   </geometry>
590   <meta>
591     <layer>0</layer>
592   </meta>
593 </visual>
594 <collision name="Wall_5_Collision_1">
595   <geometry>
596     <box>
597       <size>6.95105 0.15 2.5</size>
598     </box>
599   </geometry>
600   <pose frame="">0.524473 0 1.25 0 -0 0</pose>
601   <max_contacts>10</max_contacts>
602   <surface>
603     <contact>
604       <ode/>
605     </contact>
606     <bounce/>
607     <friction>
608       <torsional>
609         <ode/>
610       </torsional>
611         <ode/>
612     </friction>
613   </surface>
614 </collision>
615 <visual name="Wall_5_Visual_2">
616   <pose frame="">-3.40105 0 2.25 0 -0 0</pose>
617   <geometry>
618     <box>

```

```

619      <size>0.9 0.15 0.5</size>
620      </box>
621      </geometry>
622      <meta>
623        <layer>0</layer>
624      </meta>
625      </visual>
626      <collision name='Wall_5_Collision_2'>
627        <geometry>
628          <box>
629            <size>0.9 0.15 0.5</size>
630          </box>
631        </geometry>
632        <pose frame="-3.40105 0 2.25 0 -0 0"></pose>
633        <max_contacts>10</max_contacts>
634        <surface>
635          <contact>
636            <ode/>
637          </contact>
638          <bounce/>
639          <friction>
640            <torsional>
641              <ode/>
642            </torsional>
643            <ode/>
644          </friction>
645        </surface>
646      </collision>
647      <self_collide>0</self_collide>
648      <enable_wind>0</enable_wind>
649      <kinematic>0</kinematic>
650    </link>
651    <link name='Wall_6'>
652      <pose frame="-0.004434 3.925 0 0 -0 3.14159"></pose>
653      <visual name='Wall_6_Visual_0'>
654        <pose frame="-0.655307 0 1.25 0 -0 0"></pose>
655        <geometry>
656          <box>
657            <size>14.6894 0.15 2.5</size>
658          </box>
659        </geometry>
660        <meta>
661          <layer>0</layer>
662        </meta>
663      </visual>
664      <collision name='Wall_6_Collision_0'>
665        <geometry>
666          <box>
667            <size>14.6894 0.15 2.5</size>
668          </box>
669        </geometry>
670        <pose frame="-0.655307 0 1.25 0 -0 0"></pose>
671        <max_contacts>10</max_contacts>
672        <surface>
673          <contact>
674            <ode/>
675          </contact>
676          <bounce/>
677          <friction>
678            <torsional>
679              <ode/>
680            </torsional>

```

```

681      <ode/>
682      </friction>
683      </surface>
684  </collision>
685  <visual name='Wall_6_Visual_1'>
686    <pose frame="">7.79469 0 1.25 0 -0 0</pose>
687    <geometry>
688      <box>
689        <size>0.410614 0.15 2.5</size>
690      </box>
691    </geometry>
692    <meta>
693      <layer>0</layer>
694    </meta>
695  </visual>
696  <collision name='Wall_6_Collision_1'>
697    <geometry>
698      <box>
699        <size>0.410614 0.15 2.5</size>
700      </box>
701    </geometry>
702    <pose frame="">7.79469 0 1.25 0 -0 0</pose>
703    <max_contacts>10</max_contacts>
704    <surface>
705      <contact>
706        <ode/>
707      </contact>
708      <bounce/>
709      <friction>
710        <torsional>
711          <ode/>
712        </torsional>
713        <ode/>
714      </friction>
715    </surface>
716  </collision>
717  <visual name='Wall_6_Visual_2'>
718    <pose frame="">7.13939 0 2.25 0 -0 0</pose>
719    <geometry>
720      <box>
721        <size>0.9 0.15 0.5</size>
722      </box>
723    </geometry>
724    <meta>
725      <layer>0</layer>
726    </meta>
727  </visual>
728  <collision name='Wall_6_Collision_2'>
729    <geometry>
730      <box>
731        <size>0.9 0.15 0.5</size>
732      </box>
733    </geometry>
734    <pose frame="">7.13939 0 2.25 0 -0 0</pose>
735    <max_contacts>10</max_contacts>
736    <surface>
737      <contact>
738        <ode/>
739      </contact>
740      <bounce/>
741      <friction>
742        <torsional>

```

```

743      <ode/>
744      </torsional>
745      <ode/>
746      </friction>
747      </surface>
748      </collision>
749      <self_collide>0</self_collide>
750      <enable_wind>0</enable_wind>
751      <kinematic>0</kinematic>
752      </link>
753      <link name='Wall_8'>
754          <collision name='Wall_8_Collision'>
755              <geometry>
756                  <box>
757                      <size>3 0.15 2.5</size>
758                  </box>
759              </geometry>
760              <pose frame="">0 0 1.25 0 -0 0</pose>
761              <max_contacts>10</max_contacts>
762              <surface>
763                  <contact>
764                      <ode/>
765                  </contact>
766                  <bounce/>
767                  <friction>
768                      <torsional>
769                          <ode/>
770                      </torsional>
771                  <ode/>
772                  </friction>
773              </surface>
774          </collision>
775          <visual name='Wall_8_Visual'>
776              <pose frame="">0 0 1.25 0 -0 0</pose>
777              <geometry>
778                  <box>
779                      <size>3 0.15 2.5</size>
780                  </box>
781              </geometry>
782              <meta>
783                  <layer>0</layer>
784              </meta>
785          </visual>
786          <pose frame="">-3.24179 2.45506 0 0 -0 -1.5708</pose>
787          <self_collide>0</self_collide>
788          <enable_wind>0</enable_wind>
789          <kinematic>0</kinematic>
790          </link>
791          <link name='Wall_9'>
792              <collision name='Wall_9_Collision'>
793                  <geometry>
794                      <box>
795                          <size>2.5 0.15 2.5</size>
796                      </box>
797                  </geometry>
798                  <pose frame="">0 0 1.25 0 -0 0</pose>
799                  <max_contacts>10</max_contacts>
800                  <surface>
801                      <contact>
802                          <ode/>
803                      </contact>
804                      <bounce/>

```

```

805   <friction>
806     <torsional>
807       <ode/>
808     </torsional>
809   <ode/>
810   </friction>
811   </surface>
812 </collision>
813 <visual name='Wall_9_Visual'>
814   <pose frame="">0 0 1.25 0 -0 0</pose>
815   <geometry>
816     <box>
817       <size>2.5 0.15 2.5</size>
818     </box>
819   </geometry>
820   <meta>
821     <layer>0</layer>
822   </meta>
823 </visual>
824 <pose frame="">-2.06679 1.03006 0 0 -0 0</pose>
825   <self_collide>0</self_collide>
826   <enable_wind>0</enable_wind>
827   <kinematic>0</kinematic>
828 </link>
829 <static>1</static>
830 </model>
831 <state world_name='default'>
832   <sim_time>232 785000000</sim_time>
833   <real_time>118 702538223</real_time>
834   <wall_time>1551031600 831011974</wall_time>
835   <iterations>118381</iterations>
836 <model name='ground_plane'>
837   <pose frame="">0 0 0 0 -0 0</pose>
838   <scale>1 1 1</scale>
839   <link name='link'>
840     <pose frame="">0 0 0 0 -0 0</pose>
841     <velocity>0 0 0 0 -0 0</velocity>
842     <acceleration>0 0 0 0 -0 0</acceleration>
843     <wrench>0 0 0 0 -0 0</wrench>
844   </link>
845 </model>
846 <model name='sudhir.building'>
847   <pose frame="">0.673102 -0.134423 0 0 -0 0</pose>
848   <scale>1 1 1</scale>
849   <link name='Wall_0'>
850     <pose frame="">-7.25633 1.71558 0 0 0 -1.5708</pose>
851     <velocity>0 0 0 0 -0 0</velocity>
852     <acceleration>0 0 0 0 -0 0</acceleration>
853     <wrench>0 0 0 0 -0 0</wrench>
854   </link>
855   <link name='Wall_10'>
856     <pose frame="">-0.218688 -0.279363 0 0 0 -1.5708</pose>
857     <velocity>0 0 0 0 -0 0</velocity>
858     <acceleration>0 0 0 0 -0 0</acceleration>
859     <wrench>0 0 0 0 -0 0</wrench>
860   </link>
861   <link name='Wall_11'>
862     <pose frame="">-1.39369 -1.45436 0 0 -0 3.14159</pose>
863     <velocity>0 0 0 0 -0 0</velocity>
864     <acceleration>0 0 0 0 -0 0</acceleration>
865     <wrench>0 0 0 0 -0 0</wrench>
866   </link>

```

```

867 <link name="Wall_13">
868   <pose frame="">-0.235508 2.86522 0 0 0 -1.5708</pose>
869   <velocity>0 0 0 0 -0 0</velocity>
870   <acceleration>0 0 0 0 -0 0</acceleration>
871   <wrench>0 0 0 0 -0 0</wrench>
872 </link>
873 <link name="Wall_14">
874   <pose frame="">1.06449 2.06519 0 0 -0 0</pose>
875   <velocity>0 0 0 0 -0 0</velocity>
876   <acceleration>0 0 0 0 -0 0</acceleration>
877   <wrench>0 0 0 0 -0 0</wrench>
878 </link>
879 <link name="Wall_17">
880   <pose frame="">3.30448 2.88532 0 0 0 -1.5708</pose>
881   <velocity>0 0 0 0 -0 0</velocity>
882   <acceleration>0 0 0 0 -0 0</acceleration>
883   <wrench>0 0 0 0 -0 0</wrench>
884 </link>
885 <link name="Wall_18">
886   <pose frame="">4.47948 2.0853 0 0 -0 0</pose>
887   <velocity>0 0 0 0 -0 0</velocity>
888   <acceleration>0 0 0 0 -0 0</acceleration>
889   <wrench>0 0 0 0 -0 0</wrench>
890 </link>
891 <link name="Wall_20">
892   <pose frame="">6.42208 2.55052 0 0 0 -1.5708</pose>
893   <velocity>0 0 0 0 -0 0</velocity>
894   <acceleration>0 0 0 0 -0 0</acceleration>
895   <wrench>0 0 0 0 -0 0</wrench>
896 </link>
897 <link name="Wall_22">
898   <pose frame="">7.49219 0.587807 0 0 -0 0</pose>
899   <velocity>0 0 0 0 -0 0</velocity>
900   <acceleration>0 0 0 0 -0 0</acceleration>
901   <wrench>0 0 0 0 -0 0</wrench>
902 </link>
903 <link name="Wall_24">
904   <pose frame="">6.50253 -1.03685 0 0 0 -1.5708</pose>
905   <velocity>0 0 0 0 -0 0</velocity>
906   <acceleration>0 0 0 0 -0 0</acceleration>
907   <wrench>0 0 0 0 -0 0</wrench>
908 </link>
909 <link name="Wall_25">
910   <pose frame="">7.55253 -1.83685 0 0 -0 0</pose>
911   <velocity>0 0 0 0 -0 0</velocity>
912   <acceleration>0 0 0 0 -0 0</acceleration>
913   <wrench>0 0 0 0 -0 0</wrench>
914 </link>
915 <link name="Wall_3">
916   <pose frame="">-7.25633 -2.13442 0 0 0 -1.5708</pose>
917   <velocity>0 0 0 0 -0 0</velocity>
918   <acceleration>0 0 0 0 -0 0</acceleration>
919   <wrench>0 0 0 0 -0 0</wrench>
920 </link>
921 <link name="Wall_4">
922   <pose frame="">0.668672 -4.05942 0 0 -0 0</pose>
923   <velocity>0 0 0 0 -0 0</velocity>
924   <acceleration>0 0 0 0 -0 0</acceleration>
925   <wrench>0 0 0 0 -0 0</wrench>
926 </link>
927 <link name="Wall_5">
928   <pose frame="">8.59367 -0.134423 0 0 -0 1.5708</pose>

```

```

929 <velocity>0 0 0 0 -0 0</velocity>
930 <acceleration>0 0 0 0 -0 0</acceleration>
931 <wrench>0 0 0 0 -0 0</wrench>
932 </link>
933 <link name='Wall_6'>
934   <pose frame="">0.668672 3.79062 0 0 -0 3.14159</pose>
935   <velocity>0 0 0 0 -0 0</velocity>
936   <acceleration>0 0 0 0 -0 0</acceleration>
937   <wrench>0 0 0 0 -0 0</wrench>
938 </link>
939 <link name='Wall_8'>
940   <pose frame="">-2.56869 2.32064 0 0 0 -1.5708</pose>
941   <velocity>0 0 0 0 -0 0</velocity>
942   <acceleration>0 0 0 0 -0 0</acceleration>
943   <wrench>0 0 0 0 -0 0</wrench>
944 </link>
945 <link name='Wall_9'>
946   <pose frame="">-1.39369 0.895637 0 0 -0 0</pose>
947   <velocity>0 0 0 0 -0 0</velocity>
948   <acceleration>0 0 0 0 -0 0</acceleration>
949   <wrench>0 0 0 0 -0 0</wrench>
950 </link>
951 </model>
952 <light name='sun'>
953   <pose frame="">0 0 10 0 -0 0</pose>
954 </light>
955 </state>
956 <gui fullscreen='0'>
957   <camera name='user_camera'>
958     <pose frame="">15.4065 -7.32713 9.3415 -0 0.531643 2.6842</pose>
959     <view_controller>orbit</view_controller>
960     <projection_type>perspective</projection_type>
961   </camera>
962 </gui>
963 </world>
964 </sdf>

```

8. Programazio-kodea: etxea.world

7.9 gazebo.launch.py

```

1 import os
2
3 from ament_index_python.packages import get_package_share_directory
4 from launch import LaunchDescription
5 from launch.actions import DeclareLaunchArgument, ExecuteProcess, IncludeLaunchDescription
6 from launch.conditions import IfCondition
7 from launch.launch_description_sources import PythonLaunchDescriptionSource
8 from launch.substitutions import Command, LaunchConfiguration, PythonExpression
9 from launch_ros.actions import Node
10
11 def generate_launch_description():
12   # Get the launch directory
13   bringup_dir = get_package_share_directory('junki_bot')
14   launch_dir = os.path.join(bringup_dir, 'launch')
15   default_model_path = os.path.join(bringup_dir, 'deskribapena/junki_bot.urdf.xacro')
16
17   # Create the launch configuration variables
18   namespace = LaunchConfiguration('namespace')
19   use_sim_time = LaunchConfiguration('use_sim_time')
20   model = LaunchConfiguration('model')

```

```

21
22 # Launch configuration variables specific to simulation
23 use_simulator = LaunchConfiguration('use_simulator')
24 use_robot_state_pub = LaunchConfiguration('use_robot_state_pub')
25 headless = LaunchConfiguration('headless')
26 world = LaunchConfiguration('world')
27
28 # initial spawn position
29 x_pos = 0.00; y_pos = 0.00; z_pos = 0.00
30 #initial spawn orientation
31 roll = 0; pitch = 0; yaw = 0
32
33 remappings = [('/tf', 'tf'),
34               ('/tf_static', 'tf_static')]
35
36 # Declare the launch arguments
37 declare_namespace_cmd = DeclareLaunchArgument(
38   'namespace',
39   default_value='',
40   description='Top-level namespace')
41
42 declare_use_namespace_cmd = DeclareLaunchArgument(
43   'use_namespace',
44   default_value='false',
45   description='Whether to apply a namespace to the navigation stack')
46
47 declare_use_sim_time_cmd = DeclareLaunchArgument(
48   'use_sim_time',
49   default_value='true',
50   description='Use simulation (Gazebo) clock if true')
51
52 declare_use_respawn_cmd = DeclareLaunchArgument(
53   'use_respawn', default_value='False',
54   description='Whether to respawn if a node crashes. Applied when composition is disabled.')
55
56 declare_use_simulator_cmd = DeclareLaunchArgument(
57   'use_simulator',
58   default_value='True',
59   description='Whether to start the simulator')
60
61 declare_use_robot_state_pub_cmd = DeclareLaunchArgument(
62   'use_robot_state_pub',
63   default_value='True',
64   description='Whether to start the robot state publisher')
65
66 declare_simulator_cmd = DeclareLaunchArgument(
67   'headless',
68   default_value='False',
69   description='Whether to execute gzclient')
70 declare_model_path_cmd = DeclareLaunchArgument(
71   name='model',
72   default_value=default_model_path,
73   description='Absolute path to robot urdf file')
74
75 declare_world_cmd = DeclareLaunchArgument(
76   'world',
77   default_value=os.path.join(bringup_dir, 'worlds', 'etxea.world'),
78   description='Full path to world model file to load')
79
80 # Specify the actions
81 start_gazebo_server_cmd = ExecuteProcess(
82   condition=IfCondition(use_simulator),

```

```

83     cmd=['gzserver', '-s', 'libgazebo_ros_init.so',
84         '-s', 'libgazebo_ros_factory.so', world],
85     cwd=[launch_dir], output='screen')
86
87 start_gazebo_client_cmd = ExecuteProcess(
88     condition=IfCondition(PythonExpression(
89         [use_simulator, ' and not ', headless])),
90     cmd=['gzclient'],
91     cwd=[launch_dir], output='screen')
92
93 start_robot_state_publisher_cmd = Node(
94     condition=IfCondition(use_robot_state_pub),
95     package='robot_state_publisher',
96     executable='robot_state_publisher',
97     name='robot_state_publisher',
98     namespace=namespace,
99     output='screen',
100    parameters=[{'use_sim_time': use_sim_time,
101                 'robot_description': Command(['xacro ', model])}],
102    remappings=remappings,
103    arguments=[default_model_path])
104
105 start_gazebo_spawner_cmd = Node(
106     package='gazebo_ros',
107     executable='spawn_entity.py',
108     arguments=['-entity', 'junki_bot', '-topic', 'robot_description', '-x', str(x_pos), '-y', str(y_pos), '-z', str(z_pos), '-R', str(roll),
109 '-P', str(pitch), '-Y', str(yaw)],
110     output='screen')
111
112 # Create the launch description and populate
113 ld = LaunchDescription()
114
115 # Declare the launch options
116 ld.add_action(declare_namespace_cmd)
117 ld.add_action(declare_use_namespace_cmd)
118 ld.add_action(declare_use_sim_time_cmd)
119
120 ld.add_action(declare_use_simulator_cmd)
121 ld.add_action(declare_use_robot_state_pub_cmd)
122
123 ld.add_action(declare_simulator_cmd)
124 ld.add_action(declare_world_cmd)
125 ld.add_action(declare_use_respawn_cmd)
126
127 ld.add_action(declare_model_path_cmd)
128
129 # Add any conditioned actions
130 ld.add_action(start_gazebo_server_cmd)
131 ld.add_action(start_gazebo_client_cmd)
132 ld.add_action(start_gazebo_spawner_cmd)
133 # Add the actions to launch all of the navigation nodes
134 ld.add_action(start_robot_state_publisher_cmd)
135
136 return ld

```

9. Programazio-kodea: gazebo.launch.py

7.10 lidar.xacro

1 <?xml version="1.0"?>

```

2 <robot xmlns:xacro="http://www.ros.org/wiki/xacro">
3
4   <link name="lidar_link">
5     <inertial>
6       <origin xyz="0 0 0" rpy="0 0 0"/>
7       <mass value="0.125"/>
8       <inertia ixz="0.001" ixy="0" ixz="0" iyy="0.001" iyz="0" izz="0.001" />
9     </inertial>
10
11   <collision>
12     <origin xyz="0 0 0" rpy="0 0 0"/>
13     <geometry>
14       <cylinder radius="0.0508" length="0.055"/>
15     </geometry>
16   </collision>
17
18   <visual>
19     <origin xyz="0 0 0" rpy="0 0 0"/>
20     <geometry>
21       <cylinder radius="0.0508" length="0.055"/>
22     </geometry>
23   </visual>
24 </link>
25
26 <joint name="lidar_joint" type="fixed">
27   <parent link="base_link"/>
28   <child link="lidar_link"/>
29   <origin xyz="0 0 0.12" rpy="0 0 0"/>
30 </joint>
31
32 <gazebo reference="lidar_link">
33   <sensor name="lidar" type="ray">
34     <always_on>true</always_on>
35     <visualize>true</visualize>
36     <update_rate>5</update_rate>
37     <ray>
38       <scan>
39         <horizontal>
40           <samples>360</samples>
41           <resolution>1.000000</resolution>
42           <min_angle>0.000000</min_angle>
43           <max_angle>6.280000</max_angle>
44         </horizontal>
45       </scan>
46       <range>
47         <min>0.120000</min>
48         <max>3.5</max>
49         <resolution>0.015000</resolution>
50       </range>
51       <noise>
52         <type>gaussian</type>
53         <mean>0.0</mean>
54         <stddev>0.01</stddev>
55       </noise>
56     </ray>
57   <plugin name="scan" filename="libgazebo_ros_ray_sensor.so">
58     <ros>
59       <remapping>~/out:=scan</remapping>
60     </ros>
61     <output_type>sensor_msgs/LaserScan</output_type>
62     <frame_name>lidar_link</frame_name>
63   </plugin>

```

```

64      </sensor>
65    </gazebo>
66  </robot>

```

10. Programazio-kodea: lidar.xacro

7.11 nav2_params.yaml

```

1 amcl:
2   ros__parameters:
3     use_sim_time: True
4     alpha1: 0.2
5     alpha2: 0.2
6     alpha3: 0.2
7     alpha4: 0.2
8     alpha5: 0.2
9     base_frame_id: "base_footprint"
10    beam_skip_distance: 0.5
11    beam_skip_error_threshold: 0.9
12    beam_skip_threshold: 0.3
13    do_beamskip: false
14    global_frame_id: "map"
15    lambda_short: 0.1
16    laser_likelihood_max_dist: 2.0
17    laser_max_range: 100.0
18    laser_min_range: -1.0
19    laser_model_type: "likelihood_field"
20    max_beams: 60
21    max_particles: 2000
22    min_particles: 500
23    odom_frame_id: "odom"
24    pf_err: 0.05
25    pf_z: 0.99
26    recovery_alpha_fast: 0.0
27    recovery_alpha_slow: 0.0
28    resample_interval: 1
29    robot_model_type: "nav2_amcl::DifferentialMotionModel"
30    save_pose_rate: 0.5
31    sigma_hit: 0.2
32    tf_broadcast: true
33    transform_tolerance: 1.0
34    update_min_a: 0.2
35    update_min_d: 0.25
36    z_hit: 0.5
37    z_max: 0.05
38    z_rand: 0.5
39    z_short: 0.05
40    scan_topic: scan
41
42 amcl_map_client:
43   ros__parameters:
44     use_sim_time: True
45
46 amcl_rclcpp_node:
47   ros__parameters:
48     use_sim_time: True
49
50 bt_navigator:
51   ros__parameters:
52     use_sim_time: True

```

```

53 global_frame: map
54 robot_base_frame: base_link
55 odom_topic: /odom
56 bt_loop_duration: 10
57 default_server_timeout: 20
58 # 'default_nav_through_poses_bt_xml' and 'default_nav_to_pose_bt_xml' are use defaults:
59 # nav2_bt_navigator/navigate_to_pose_w_replanning_and_recovery.xml
60 # nav2_bt_navigator/navigate_through_poses_w_replanning_and_recovery.xml
61 # They can be set here or via a RewrittenYaml remap from a parent launch file to Nav2.
62 plugin_lib_names:
63 - nav2_compute_path_to_pose_action_bt_node
64 - nav2_compute_path_through_poses_action_bt_node
65 - nav2_smooth_path_action_bt_node
66 - nav2_follow_path_action_bt_node
67 - nav2_spin_action_bt_node
68 - nav2_wait_action_bt_node
69 - nav2_back_up_action_bt_node
70 - nav2_drive_on_heading_bt_node
71 - nav2_clear_costmap_service_bt_node
72 - nav2_is_stuck_condition_bt_node
73 - nav2_goal_reached_condition_bt_node
74 - nav2_goal_updated_condition_bt_node
75 - nav2_globally_updated_goal_condition_bt_node
76 - nav2_is_path_valid_condition_bt_node
77 - nav2_initial_pose_received_condition_bt_node
78 - nav2_reinitialize_global_localization_service_bt_node
79 - nav2_rate_controller_bt_node
80 - nav2_distance_controller_bt_node
81 - nav2_speed_controller_bt_node
82 - nav2_truncate_path_action_bt_node
83 - nav2_truncate_path_local_action_bt_node
84 - nav2_goal_updater_node_bt_node
85 - nav2_recovery_node_bt_node
86 - nav2_pipeline_sequence_bt_node
87 - nav2_round_robin_node_bt_node
88 - nav2_transform_available_condition_bt_node
89 - nav2_time_expired_condition_bt_node
90 - nav2_path_expiring_timer_condition
91 - nav2_distance_traveled_condition_bt_node
92 - nav2_single_trigger_bt_node
93 - nav2_goal_updated_controller_bt_node
94 - nav2_is_battery_low_condition_bt_node
95 - nav2_navigate_through_poses_action_bt_node
96 - nav2_navigate_to_pose_action_bt_node
97 - nav2_remove_passed_goals_action_bt_node
98 - nav2_planner_selector_bt_node
99 - nav2_controller_selector_bt_node
100 - nav2_goal_checker_selector_bt_node
101 - nav2_controller_cancel_bt_node
102 - nav2_path_longer_on_approach_bt_node
103 - nav2_wait_cancel_bt_node
104 - nav2_spin_cancel_bt_node
105 - nav2_back_up_cancel_bt_node
106 - nav2_drive_on_heading_cancel_bt_node
107
108 bt_navigator_rclcpp_node:
109   ros__parameters:
110     use_sim_time: True
111
112 controller_server:
113   ros__parameters:
114     use_sim_time: True

```

```

115 controller_frequency: 20.0
116 min_x_velocity_threshold: 0.001
117 min_y_velocity_threshold: 0.5
118 min_theta_velocity_threshold: 0.001
119 failure_tolerance: 0.3
120 progress_checker_plugin: "progress_checker"
121 goal_checker_plugins: ["general_goal_checker"] # "precise_goal_checker"
122 controller_plugins: ["FollowPath"]
123
124 # Progress checker parameters
125 progress_checker:
126   plugin: "nav2_controller::SimpleProgressChecker"
127   required_movement_radius: 0.5
128   movement_time_allowance: 10.0
129 # Goal checker parameters
130 #precise_goal_checker:
131   # plugin: "nav2_controller::SimpleGoalChecker"
132   # xy_goal_tolerance: 0.25
133   # yaw_goal_tolerance: 0.25
134   # stateful: True
135 general_goal_checker:
136   stateful: True
137   plugin: "nav2_controller::SimpleGoalChecker"
138   xy_goal_tolerance: 0.25
139   yaw_goal_tolerance: 0.25
140 # DWB parameters
141 FollowPath:
142   plugin: "dwb_core::DWBLocalPlanner"
143   debug_trajectory_details: True
144   min_vel_x: 0.0
145   min_vel_y: 0.0
146   max_vel_x: 0.26
147   max_vel_y: 0.0
148   max_vel_theta: 1.0
149   min_speed_xy: 0.0
150   max_speed_xy: 0.26
151   min_speed_theta: 0.0
152 # Add high threshold velocity for turtlebot 3 issue.
153 # https://github.com/ROBOTIS-GIT/turtlebot3\_simulations/issues/75
154 acc_lim_x: 2.5
155 acc_lim_y: 0.0
156 acc_lim_theta: 3.2
157 decel_lim_x: -2.5
158 decel_lim_y: 0.0
159 decel_lim_theta: -3.2
160 vx_samples: 20
161 vy_samples: 5
162 vtheta_samples: 20
163 sim_time: 1.7
164 linear_granularity: 0.05
165 angular_granularity: 0.025
166 transform_tolerance: 0.2
167 xy_goal_tolerance: 0.25
168 trans_stopped_velocity: 0.25
169 short_circuit_trajectory_evaluation: True
170 stateful: True
171 critics: ["RotateToGoal", "Oscillation", "BaseObstacle", "GoalAlign", "PathAlign", "PathDist", "GoalDist"]
172 BaseObstacle.scale: 0.02
173 PathAlign.scale: 32.0
174 PathAlign.forward_point_distance: 0.1
175 GoalAlign.scale: 24.0
176 GoalAlign.forward_point_distance: 0.1

```

```

177 PathDist.scale: 32.0
178 GoalDist.scale: 24.0
179 RotateToGoal.scale: 32.0
180 RotateToGoal.slowing_factor: 5.0
181 RotateToGoal.lookahead_time: -1.0
182
183 controller_server_rclcpp_node:
184   ros__parameters:
185     use_sim_time: True
186
187 local_costmap:
188   local_costmap:
189     ros__parameters:
190       update_frequency: 5.0
191       publish_frequency: 2.0
192       global_frame: odom
193       robot_base_frame: base_link
194       use_sim_time: True
195       rolling_window: true
196       width: 3
197       height: 3
198       resolution: 0.05
199       robot_radius: 0.22
200       plugins: ["voxel_layer", "inflation_layer"]
201       inflation_layer:
202         plugin: "nav2_costmap_2d::InflationLayer"
203         cost_scaling_factor: 3.0
204         inflation_radius: 0.55
205       voxel_layer:
206         plugin: "nav2_costmap_2d::VoxelLayer"
207         enabled: True
208         publish_voxel_map: True
209         origin_z: 0.0
210         z_resolution: 0.05
211         z_voxels: 16
212         max_obstacle_height: 2.0
213         mark_threshold: 0
214         observation_sources: scan
215         scan:
216           topic: /scan
217           max_obstacle_height: 2.0
218           clearing: True
219           marking: True
220           data_type: "LaserScan"
221           raytrace_max_range: 3.0
222           raytrace_min_range: 0.0
223           obstacle_max_range: 2.5
224           obstacle_min_range: 0.0
225       static_layer:
226         map_subscribe_transient_local: True
227         always_send_full_costmap: True
228   local_costmap_client:
229     ros__parameters:
230       use_sim_time: True
231   local_costmap_rclcpp_node:
232     ros__parameters:
233       use_sim_time: True
234
235 global_costmap:
236   global_costmap:
237     ros__parameters:
238       update_frequency: 1.0
  
```

```

239   publish_frequency: 1.0
240   global_frame: map
241   robot_base_frame: base_link
242   use_sim_time: True
243   robot_radius: 0.22
244   resolution: 0.05
245   track_unknown_space: true
246   plugins: ["static_layer", "obstacle_layer", "inflation_layer"]
247   obstacle_layer:
248     plugin: "nav2_costmap_2d::ObstacleLayer"
249     enabled: True
250     observation_sources: scan
251     scan:
252       topic: /scan
253       max_obstacle_height: 2.0
254       clearing: True
255       marking: True
256       data_type: "LaserScan"
257       raytrace_max_range: 3.0
258       raytrace_min_range: 0.0
259       obstacle_max_range: 2.5
260       obstacle_min_range: 0.0
261   static_layer:
262     plugin: "nav2_costmap_2d::StaticLayer"
263     map_subscribe_transient_local: True
264   inflation_layer:
265     plugin: "nav2_costmap_2d::InflationLayer"
266     cost_scaling_factor: 3.0
267     inflation_radius: 0.55
268     always_send_full_costmap: True
269   global_costmap_client:
270     ros_parameters:
271       use_sim_time: True
272   global_costmap_rclcpp_node:
273     ros_parameters:
274       use_sim_time: True
275
276   map_server:
277     ros_parameters:
278       use_sim_time: True
279     yaml_filename: "turtlebot3_world.yaml"
280
281   map_saver:
282     ros_parameters:
283       use_sim_time: True
284       save_map_timeout: 5.0
285       free_thresh_default: 0.25
286       occupied_thresh_default: 0.65
287     map_subscribe_transient_local: True
288
289   planner_server:
290     ros_parameters:
291       expected_planner_frequency: 20.0
292       use_sim_time: True
293     planner_plugins: ["GridBased"]
294     GridBased:
295       plugin: "nav2_navfn_planner/NavfnPlanner"
296       tolerance: 0.5
297       use_astar: false
298       allow_unknown: true
299
300   planner_server_rclcpp_node:
  
```

```

301 ros__parameters:
302   use_sim_time: True
303
304 smoother_server:
305   ros__parameters:
306     use_sim_time: True
307     smoother_plugins: ["simple_smoothen"]
308     simple_smoothen:
309       plugin: "nav2_smoothen::SimpleSmoothen"
310       tolerance: 1.0e-10
311       max_its: 1000
312       do_refinement: True
313
314 behavior_server:
315   ros__parameters:
316     costmap_topic: local_costmap/costmap_raw
317     footprint_topic: local_costmap/published_footprint
318     cycle_frequency: 10.0
319     behavior_plugins: ["spin", "backup", "drive_on_heading", "wait"]
320     spin:
321       plugin: "nav2_behaviors/Spin"
322     backup:
323       plugin: "nav2_behaviors/BackUp"
324     drive_on_heading:
325       plugin: "nav2_behaviors/DriveOnHeading"
326     wait:
327       plugin: "nav2_behaviors/Wait"
328     global_frame: odom
329     robot_base_frame: base_link
330     transform_tolerance: 0.1
331     use_sim_time: true
332     simulate_ahead_time: 2.0
333     max_rotational_vel: 1.0
334     min_rotational_vel: 0.4
335     rotational_acc_lim: 3.2
336
337 robot_state_publisher:
338   ros__parameters:
339     use_sim_time: True
340
341 waypoint_follower:
342   ros__parameters:
343     loop_rate: 20
344     stop_on_failure: false
345     waypoint_task_executor_plugin: "wait_at_waypoint"
346     wait_at_waypoint:
347       plugin: "nav2_waypoint_follower::WaitAtWaypoint"
348       enabled: True
349       waypoint_pause_duration: 200
  
```

11. Programazio-kodea: nav2_params.yaml

7.12 nav2_default_view.rviz

```

1 Panels:
2   - Class: rviz_common/Displays
3     Help Height: 0
4     Name: Displays
5     Property Tree Widget:
6       Expanded:
  
```

```

7   - /Global Options1
8   - /TF1/Frames1
9   - /TF1/Tree1
10  Splitter Ratio: 0.5833333134651184
11  Tree Height: 606
12  - Class: rviz_common/Selection
13  Name: Selection
14  - Class: rviz_common/Tool Properties
15  Expanded:
16    - /Publish Point1
17  Name: Tool Properties
18  Splitter Ratio: 0.5886790156364441
19  - Class: rviz_common/Views
20  Expanded:
21    - /Current View1
22  Name: Views
23  Splitter Ratio: 0.5
24  - Class: nav2_rviz_plugins/Navigation 2
25  Name: Navigation 2
26  Visualization Manager:
27  Class: ""
28  Displays:
29    - Alpha: 0.5
30    Cell Size: 1
31    Class: rviz_default_plugins/Grid
32    Color: 160; 160; 164
33    Enabled: true
34    Line Style:
35      Line Width: 0.02999999329447746
36      Value: Lines
37    Name: Grid
38    Normal Cell Count: 0
39    Offset:
40      X: 0
41      Y: 0
42      Z: 0
43    Plane: XY
44    Plane Cell Count: 10
45    Reference Frame: <Fixed Frame>
46    Value: true
47    - Alpha: 1
48    Class: rviz_default_plugins/RobotModel
49    Collision Enabled: false
50    Description File: ""
51    Description Source: Topic
52    Description Topic:
53      Depth: 5
54      Durability Policy: Volatile
55      History Policy: Keep Last
56      Reliability Policy: Reliable
57      Value: /robot_description
58    Enabled: false
59    Links:
60      All Links Enabled: true
61      Expand Joint Details: false
62      Expand Link Details: false
63      Expand Tree: false
64      Link Tree Style: ""
65    Mass Properties:
66      Inertia: false
67      Mass: false
68    Name: RobotModel
  
```

```
69    TF Prefix: ""
70    Update Interval: 0
71    Value: false
72    Visual Enabled: true
73    - Class: rviz_default_plugins/TF
74        Enabled: true
75        Frame Timeout: 15
76        Frames:
77            All Enabled: false
78            base_footprint:
79                Value: true
80            base_link:
81                Value: true
82            base_scan:
83                Value: true
84            camera_depth_frame:
85                Value: true
86            camera_depth_optical_frame:
87                Value: true
88            camera_link:
89                Value: true
90            camera_rgb_frame:
91                Value: true
92            camera_rgb_optical_frame:
93                Value: true
94            aurrekoezk_gurp_link:
95                Value: true
96            aurrekoesk_gurp_link:
97                Value: true
98            imu_link:
99                Value: true
100           map:
101               Value: true
102           odom:
103               Value: true
104           atzekoezk_gurp_link:
105               Value: true
106           atzekoesk_gurp_link:
107               Value: true
108           lidar_link:
109               Value: true
110           Marker Scale: 1
111           Name: TF
112           Show Arrows: true
113           Show Axes: true
114           Show Names: false
115           Tree:
116               map:
117               odom:
118                   base_footprint:
119                   base_link:
120                   base_scan:
121                       {}
122                   camera_link:
123                       camera_depth_frame:
124                       camera_depth_optical_frame:
125                           {}
126                   camera_rgb_frame:
127                       camera_rgb_optical_frame:
128                           {}
129                   aurrekoezk_gurp_link:
130                       {}
```

```

131      aurrekoesk_gurp_link:
132          {}
133      imu_link:
134          {}
135      lidar_link:
136          {}
137      atzekoezk_gurp_link:
138          {}
139      atzekoesk_gurp_link:
140          {}
141  Update Interval: 0
142  Value: true
143 - Alpha: 1
144  Autocompute Intensity Bounds: true
145  Autocompute Value Bounds:
146      Max Value: 10
147      Min Value: -10
148      Value: true
149  Axis: Z
150  Channel Name: intensity
151  Class: rviz_default_plugins/LaserScan
152  Color: 255; 255; 255
153  Color Transformer: Intensity
154  Decay Time: 0
155  Enabled: true
156  Invert Rainbow: false
157  Max Color: 255; 255; 255
158  Max Intensity: 0
159  Min Color: 0; 0; 0
160  Min Intensity: 0
161  Name: LaserScan
162  Position Transformer: XYZ
163  Selectable: true
164  Size (Pixels): 3
165  Size (m): 0.009999999776482582
166  Style: Points
167  Topic:
168  Depth: 5
169  Durability Policy: Volatile
170  Filter size: 10
171  History Policy: Keep Last
172  Reliability Policy: Best Effort
173  Value: /scan
174  Use Fixed Frame: true
175  Use rainbow: true
176  Value: true
177 - Alpha: 1
178  Autocompute Intensity Bounds: true
179  Autocompute Value Bounds:
180      Max Value: 10
181      Min Value: -10
182      Value: true
183  Axis: Z
184  Channel Name: intensity
185  Class: rviz_default_plugins/PointCloud2
186  Color: 255; 255; 255
187  Color Transformer: ""
188  Decay Time: 0
189  Enabled: true
190  Invert Rainbow: false
191  Max Color: 255; 255; 255
192  Max Intensity: 4096
  
```

```

193   Min Color: 0; 0; 0
194   Min Intensity: 0
195   Name: Bumper Hit
196   Position Transformer: ""
197   Selectable: true
198   Size (Pixels): 3
199   Size (m): 0.07999999821186066
200   Style: Spheres
201   Topic:
202     Depth: 5
203     Durability Policy: Volatile
204     Filter size: 10
205     History Policy: Keep Last
206     Reliability Policy: Best Effort
207     Value: /mobile_base/sensors/bumper_pointcloud
208   Use Fixed Frame: true
209   Use rainbow: true
210   Value: true
211     - Alpha: 1
212     Class: rviz_default_plugins/Map
213     Color Scheme: map
214     Draw Behind: true
215     Enabled: true
216     Name: Map
217     Topic:
218       Depth: 1
219       Durability Policy: Transient Local
220       History Policy: Keep Last
221       Reliability Policy: Reliable
222       Value: /map
223     Use Timestamp: false
224     Value: true
225     - Alpha: 1
226     Class: nav2_rviz_plugins/ParticleCloud
227     Color: 0; 180; 0
228     Enabled: true
229     Max Arrow Length: 0.3
230     Min Arrow Length: 0.02
231     Name: Amcl Particle Swarm
232     Shape: Arrow (Flat)
233     Topic:
234       Depth: 5
235       Durability Policy: Volatile
236       History Policy: Keep Last
237       Reliability Policy: Best Effort
238       Value: /particle_cloud
239     Value: true
240     - Class: rviz_common/Group
241     Displays:
242       - Alpha: 0.3
243       Class: rviz_default_plugins/Map
244       Color Scheme: costmap
245       Draw Behind: false
246       Enabled: true
247       Name: Global Costmap
248       Topic:
249         Depth: 1
250         Durability Policy: Transient Local
251         History Policy: Keep Last
252         Reliability Policy: Reliable
253         Value: /global_costmap/costmap
254     Use Timestamp: false
  
```

```

255   Value: true
256 - Alpha: 0.3
257 Class: rviz_default_plugins/Map
258 Color Scheme: costmap
259 Draw Behind: false
260 Enabled: true
261 Name: Downsampled Costmap
262 Topic:
263   Depth: 1
264   Durability Policy: Transient Local
265   Filter size: 10
266   History Policy: Keep Last
267   Reliability Policy: Reliable
268   Value: /downsampled_costmap
269 Update Topic:
270   Depth: 5
271   Durability Policy: Volatile
272   History Policy: Keep Last
273   Reliability Policy: Reliable
274   Value: /downsampled_costmap_updates
275 Use Timestamp: false
276 Value: true
277 - Alpha: 1
278 Buffer Length: 1
279 Class: rviz_default_plugins/Path
280 Color: 255; 0; 0
281 Enabled: true
282 Head Diameter: 0.01999999552965164
283 Head Length: 0.01999999552965164
284 Length: 0.30000001192092896
285 Line Style: Lines
286 Line Width: 0.02999999329447746
287 Name: Path
288 Offset:
289   X: 0
290   Y: 0
291   Z: 0
292 Pose Color: 255; 85; 255
293 Pose Style: Arrows
294 Radius: 0.02999999329447746
295 Shaft Diameter: 0.00499999888241291
296 Shaft Length: 0.01999999552965164
297 Topic:
298   Depth: 5
299   Durability Policy: Volatile
300   Filter size: 10
301   History Policy: Keep Last
302   Reliability Policy: Reliable
303   Value: /plan
304 Value: true
305 - Alpha: 1
306 Autocompute Intensity Bounds: true
307 Autocompute Value Bounds:
308   Max Value: 10
309   Min Value: -10
310   Value: true
311 Axis: Z
312 Channel Name: intensity
313 Class: rviz_default_plugins/PointCloud2
314 Color: 125; 125; 125
315 Color Transformer: FlatColor
316 Decay Time: 0
  
```

```

317   Enabled: true
318   Invert Rainbow: false
319   Max Color: 255; 255; 255
320   Max Intensity: 4096
321   Min Color: 0; 0; 0
322   Min Intensity: 0
323   Name: VoxelGrid
324   Position Transformer: XYZ
325   Selectable: true
326   Size (Pixels): 3
327   Size (m): 0.05000000074505806
328   Style: Boxes
329   Topic:
330     Depth: 5
331     Durability Policy: Volatile
332     Filter size: 10
333     History Policy: Keep Last
334     Reliability Policy: Reliable
335     Value: /global_costmap/voxel_marked_cloud
336   Use Fixed Frame: true
337   Use rainbow: true
338   Value: true
339   - Alpha: 1
340   Class: rviz_default_plugins/Polygon
341   Color: 25; 255; 0
342   Enabled: false
343   Name: Polygon
344   Topic:
345     Depth: 5
346     Durability Policy: Volatile
347     Filter size: 10
348     History Policy: Keep Last
349     Reliability Policy: Reliable
350     Value: /global_costmap/published_footprint
351   Value: false
352   Enabled: true
353   Name: Global Planner
354   - Class: rviz_common/Group
355   Displays:
356     - Alpha: 0.699999988079071
357     Class: rviz_default_plugins/Map
358     Color Scheme: costmap
359     Draw Behind: false
360     Enabled: true
361     Name: Local Costmap
362     Topic:
363       Depth: 1
364       Durability Policy: Transient Local
365       Filter size: 10
366       History Policy: Keep Last
367       Reliability Policy: Reliable
368       Value: /local_costmap/costmap
369     Update Topic:
370       Depth: 5
371       Durability Policy: Volatile
372       History Policy: Keep Last
373       Reliability Policy: Reliable
374       Value: /local_costmap/costmap_updates
375     Use Timestamp: false
376     Value: true
377   - Alpha: 1
378   Buffer Length: 1
  
```

```

379   Class: rviz_default_plugins/Path
380   Color: 0; 12; 255
381   Enabled: true
382   Head Diameter: 0.30000001192092896
383   Head Length: 0.20000000298023224
384   Length: 0.30000001192092896
385   Line Style: Lines
386   Line Width: 0.029999999329447746
387   Name: Local Plan
388   Offset:
389     X: 0
390     Y: 0
391     Z: 0
392   Pose Color: 255; 85; 255
393   Pose Style: None
394   Radius: 0.029999999329447746
395   Shaft Diameter: 0.10000000149011612
396   Shaft Length: 0.10000000149011612
397   Topic:
398     Depth: 5
399     Durability Policy: Volatile
400     Filter size: 10
401     History Policy: Keep Last
402     Reliability Policy: Reliable
403     Value: /local_plan
404     Value: true
405   - Class: rviz_default_plugins/MarkerArray
406     Enabled: false
407     Name: Trajectories
408     Namespaces:
409       {}
410     Topic:
411       Depth: 5
412       Durability Policy: Volatile
413       History Policy: Keep Last
414       Reliability Policy: Reliable
415       Value: /marker
416       Value: false
417     - Alpha: 1
418       Class: rviz_default_plugins/Polygon
419       Color: 25; 255; 0
420       Enabled: true
421       Name: Polygon
422       Topic:
423         Depth: 5
424         Durability Policy: Volatile
425         Filter size: 10
426         History Policy: Keep Last
427         Reliability Policy: Reliable
428         Value: /local_costmap/published_footprint
429         Value: true
430     - Alpha: 1
431       Autocompute Intensity Bounds: true
432       Autocompute Value Bounds:
433         Max Value: 10
434         Min Value: -10
435         Value: true
436       Axis: Z
437       Channel Name: intensity
438       Class: rviz_default_plugins/PointCloud2
439       Color: 255; 255; 255
440       Color Transformer: RGB8
  
```

```

441 Decay Time: 0
442 Enabled: true
443 Invert Rainbow: false
444 Max Color: 255; 255; 255
445 Max Intensity: 4096
446 Min Color: 0; 0; 0
447 Min Intensity: 0
448 Name: VoxelGrid
449 Position Transformer: XYZ
450 Selectable: true
451 Size (Pixels): 3
452 Size (m): 0.009999999776482582
453 Style: Flat Squares
454 Topic:
455   Depth: 5
456   Durability Policy: Volatile
457   Filter size: 10
458   History Policy: Keep Last
459   Reliability Policy: Reliable
460   Value: /local_costmap/voxel_marked_cloud
461 Use Fixed Frame: true
462 Use rainbow: true
463 Value: true
464 Enabled: true
465 Name: Controller
466 - Class: rviz_common/Group
467 Displays:
468   - Class: rviz_default_plugins/Image
469     Enabled: true
470     Max Value: 1
471     Median window: 5
472     Min Value: 0
473     Name: RealsenseCamera
474     Normalize Range: true
475     Topic:
476       Depth: 5
477       Durability Policy: Volatile
478       History Policy: Keep Last
479       Reliability Policy: Reliable
480       Value: /intel_realsense_r200_depth/image_raw
481     Value: true
482   - Alpha: 1
483     Autocompute Intensity Bounds: true
484     Autocompute Value Bounds:
485       Max Value: 10
486       Min Value: -10
487     Value: true
488     Axis: Z
489     Channel Name: intensity
490     Class: rviz_default_plugins/PointCloud2
491     Color: 255; 255; 255
492     Color Transformer: RGB8
493     Decay Time: 0
494     Enabled: true
495     Invert Rainbow: false
496     Max Color: 255; 255; 255
497     Max Intensity: 4096
498     Min Color: 0; 0; 0
499     Min Intensity: 0
500     Name: RealsenseDepthImage
501     Position Transformer: XYZ
502     Selectable: true
  
```

```

503     Size (Pixels): 3
504     Size (m): 0.009999999776482582
505     Style: Flat Squares
506     Topic:
507       Depth: 5
508       Durability Policy: Volatile
509       Filter size: 10
510       History Policy: Keep Last
511       Reliability Policy: Reliable
512       Value: /intel_realsense_r200_depth/points
513     Use Fixed Frame: true
514     Use rainbow: true
515     Value: true
516   Enabled: false
517   Name: Realsense
518 - Class: rviz_default_plugins/MarkerArray
519   Enabled: true
520   Name: MarkerArray
521   Namespaces:
522   {}
523   Topic:
524     Depth: 5
525     Durability Policy: Volatile
526     History Policy: Keep Last
527     Reliability Policy: Reliable
528     Value: /waypoints
529     Value: true
530   Enabled: true
531   Global Options:
532     Background Color: 48; 48; 48
533     Fixed Frame: map
534     Frame Rate: 30
535   Name: root
536   Tools:
537 - Class: rviz_default_plugins/MoveCamera
538 - Class: rviz_default_plugins/Select
539 - Class: rviz_default_plugins/FocusCamera
540 - Class: rviz_default_plugins/Measure
541   Line color: 128; 128; 0
542 - Class: rviz_default_plugins/SetInitialPose
543   Covariance x: 0.25
544   Covariance y: 0.25
545   Covariance yaw: 0.06853891909122467
546   Topic:
547     Depth: 5
548     Durability Policy: Volatile
549     History Policy: Keep Last
550     Reliability Policy: Reliable
551     Value: /initialpose
552 - Class: rviz_default_plugins/PublishPoint
553   Single click: true
554   Topic:
555     Depth: 5
556     Durability Policy: Volatile
557     History Policy: Keep Last
558     Reliability Policy: Reliable
559     Value: /clicked_point
560 - Class: nav2_rviz_plugins/GoalTool
561 Transformation:
562   Current:
563     Class: rviz_default_plugins/TF
564   Value: true
  
```


7.13 rviz_launch.py

```

1 import os
2
3 from ament_index_python.packages import get_package_share_directory
4
5 from launch import LaunchDescription
6 from launch.actions import DeclareLaunchArgument, EmitEvent, RegisterEventHandler
7 from launch.conditions import IfCondition, UnlessCondition
8 from launch.event_handlers import OnProcessExit
9 from launch.events import Shutdown
10 from launch.substitutions import LaunchConfiguration
11 from launch_ros.actions import Node
12 from nav2_common.launch import ReplaceString
13
14
15 def generate_launch_description():
16   # Get the launch directory
17   bringup_dir = get_package_share_directory('junki_bot')
18
19   # Create the launch configuration variables
20   namespace = LaunchConfiguration('namespace')
21   use_namespace = LaunchConfiguration('use_namespace')
22   rviz_config_file = LaunchConfiguration('rviz_config')
23
24   # Declare the launch arguments
25   declare_namespace_cmd = DeclareLaunchArgument(
26     'namespace',
27     default_value='navigation',
28     description=('Top-level namespace. The value will be used to replace the '
29                 '<robot_namespace> keyword on the rviz config file.'))
30
31   declare_use_namespace_cmd = DeclareLaunchArgument(
32     'use_namespace',
33     default_value='false',
34     description='Whether to apply a namespace to the navigation stack')
35
36   declare_rviz_config_file_cmd = DeclareLaunchArgument(
37     'rviz_config',
38     default_value=os.path.join(bringup_dir, 'rviz', 'nav2_default_view.rviz'),
39     description='Full path to the RVIZ config file to use')
40
41   # Launch rviz
42   start_rviz_cmd = Node(
43     condition=UnlessCondition(use_namespace),
44     package='rviz2',
45     executable='rviz2',
46     arguments=['-d', rviz_config_file],
47     output='screen')
48
49   namespaced_rviz_config_file = ReplaceString(
50     source_file=rviz_config_file,
51     replacements={'<robot_namespace>': ('/', namespace)})
52
53   start_namespaced_rviz_cmd = Node(
54     condition=IfCondition(use_namespace),
55     package='rviz2',
56     executable='rviz2',
57     namespace=namespace,
58     arguments=['-d', namespaced_rviz_config_file],
59     output='screen',

```

```

60     remappings=[('/tf', 'tf'),
61         ('/tf_static', 'tf_static'),
62         ('/goal_pose', 'goal_pose'),
63         ('/clicked_point', 'clicked_point'),
64         ('/initialpose', 'initialpose'))]
65
66     exit_event_handler = RegisterEventHandler(
67         condition=UnlessCondition(use_namespace),
68         event_handler=OnProcessExit(
69             target_action=start_rviz_cmd,
70             on_exit=EmitEvent(event=Shutdown(reason='rviz exited'))))
71
72     exit_event_handler_namespaced = RegisterEventHandler(
73         condition=IfCondition(use_namespace),
74         event_handler=OnProcessExit(
75             target_action=start_namespaced_rviz_cmd,
76             on_exit=EmitEvent(event=Shutdown(reason='rviz exited'))))
77
78     # Create the launch description and populate
79     ld = LaunchDescription()
80
81     # Declare the launch options
82     ld.add_action(declare_namespace_cmd)
83     ld.add_action(declare_use_namespace_cmd)
84     ld.add_action(declare_rviz_config_file_cmd)
85
86     # Add any conditioned actions
87     ld.add_action(start_rviz_cmd)
88     ld.add_action(start_namespaced_rviz_cmd)
89
90     # Add other nodes and processes we need
91     ld.add_action(exit_event_handler)
92     ld.add_action(exit_event_handler_namespaced)
93
94     return ld

```

13. Programazio-kodea: rviz_launch.py

7.14 nabegazioa.launch.py

```

1 import os
2
3 from ament_index_python.packages import get_package_share_directory
4
5 from launch import LaunchDescription
6 from launch.actions import DeclareLaunchArgument, ExecuteProcess, IncludeLaunchDescription
7 from launch.conditions import IfCondition
8 from launch.launch_description_sources import PythonLaunchDescriptionSource
9 from launch.substitutions import LaunchConfiguration, PythonExpression
10 from launch_ros.actions import Node
11 from launch_ros.substitutions import FindPackageShare
12
13 def generate_launch_description():
14     # Get the launch directory
15     bringup_dir = get_package_share_directory('junki_bot')
16     launch_dir = os.path.join(bringup_dir, 'launch')
17
18     nav2_dir = FindPackageShare(package='nav2_bringup').find('nav2_bringup')
19     nav2_launch_dir = os.path.join(nav2_dir, 'launch')
20

```

```

21 # Create the launch configuration variables
22 slam = LaunchConfiguration('slam')
23 namespace = LaunchConfiguration('namespace')
24 use_namespace = LaunchConfiguration('use_namespace')
25 map_yaml_file = LaunchConfiguration('map')
26 use_sim_time = LaunchConfiguration('use_sim_time')
27 params_file = LaunchConfiguration('params_file')
28 autostart = LaunchConfiguration('autostart')
29 use_composition = LaunchConfiguration('use_composition')
30 use_respawn = LaunchConfiguration('use_respawn')
31
32 # Launch configuration variables specific to simulation
33 rviz_config_file = LaunchConfiguration('rviz_config_file')
34 use_rviz = LaunchConfiguration('use_rviz')
35
36 # Declare the launch arguments
37 declare_namespace_cmd = DeclareLaunchArgument(
38   'namespace',
39   default_value='',
40   description='Top-level namespace')
41
42 declare_use_namespace_cmd = DeclareLaunchArgument(
43   'use_namespace',
44   default_value='false',
45   description='Whether to apply a namespace to the navigation stack')
46
47 declare_slam_cmd = DeclareLaunchArgument(
48   'slam',
49   default_value='True',
50   description='Whether run a SLAM')
51
52 declare_map_yaml_cmd = DeclareLaunchArgument(
53   'map',
54   default_value=os.path.join(
55     bringup_dir, 'maps', 'etxea_mapa.yaml'),
56   description='Full path to map file to load')
57
58 declare_use_sim_time_cmd = DeclareLaunchArgument(
59   'use_sim_time',
60   default_value='true',
61   description='Use simulation (Gazebo) clock if true')
62
63 declare_params_file_cmd = DeclareLaunchArgument(
64   'params_file',
65   default_value=os.path.join(bringup_dir, 'params', 'nav2_params.yaml'),
66   description='Full path to the ROS2 parameters file to use for all launched nodes')
67
68 declare_autostart_cmd = DeclareLaunchArgument(
69   'autostart', default_value='true',
70   description='Automatically startup the nav2 stack')
71
72 declare_use_composition_cmd = DeclareLaunchArgument(
73   'use_composition', default_value='True',
74   description='Whether to use composed bringup')
75
76 declare_use_respawn_cmd = DeclareLaunchArgument(
77   'use_respawn', default_value='False',
78   description='Whether to respawn if a node crashes. Applied when composition is disabled.')
79
80 declare_rviz_config_file_cmd = DeclareLaunchArgument(
81   'rviz_config_file',
82   default_value=os.path.join(

```

```

83     bringup_dir, 'rviz', 'nav2_default_view.rviz'),
84     description='Full path to the RVIZ config file to use')
85
86 declare_use_rviz_cmd = DeclareLaunchArgument(
87   'use_rviz',
88   default_value='True',
89   description='Whether to start RVIZ')
90
91 # Specify the actions
92 urdf = os.path.join(bringup_dir, 'deskribapena', 'junki_bot.urdf.xacro')
93 with open(urdf, 'r') as infp:
94   robot_description = infp.read()
95
96 rviz_cmd = IncludeLaunchDescription(
97   PythonLaunchDescriptionSource(
98     os.path.join(launch_dir, 'rviz_launch.py')),
99   condition=IfCondition(use_rviz),
100  launch_arguments={'namespace': namespace,
101    'use_namespace': use_namespace,
102    'rviz_config': rviz_config_file}.items())
103
104 bringup_cmd = IncludeLaunchDescription(
105   PythonLaunchDescriptionSource(
106     os.path.join(nav2_launch_dir, 'bringup.launch.py')),
107   launch_arguments={'namespace': namespace,
108     'use_namespace': use_namespace,
109     'slam': slam,
110     'map': map_yaml_file,
111     'use_sim_time': use_sim_time,
112     'params_file': params_file,
113     'autostart': autostart,
114     'use_composition': use_composition,
115     'use_respawn': use_respawn}.items())
116
117 # Create the launch description and populate
118 ld = LaunchDescription()
119
120 # Declare the launch options
121 ld.add_action(declare_namespace_cmd)
122 ld.add_action(declare_use_namespace_cmd)
123 ld.add_action(declare_slam_cmd)
124 ld.add_action(declare_map_yaml_cmd)
125 ld.add_action(declare_use_sim_time_cmd)
126 ld.add_action(declare_params_file_cmd)
127 ld.add_action(declare_autostart_cmd)
128 ld.add_action(declare_use_composition_cmd)
129
130 ld.add_action(declare_rviz_config_file_cmd)
131 ld.add_action(declare_use_rviz_cmd)
132
133 ld.add_action(declare_use_respawn_cmd)
134
135 # Add the actions to launch all of the navigation nodes
136 ld.add_action(rviz_cmd)
137 ld.add_action(bringup_cmd)
138
139 return ld

```

14. Programazio-kodea: nabegazioa.launch.py