

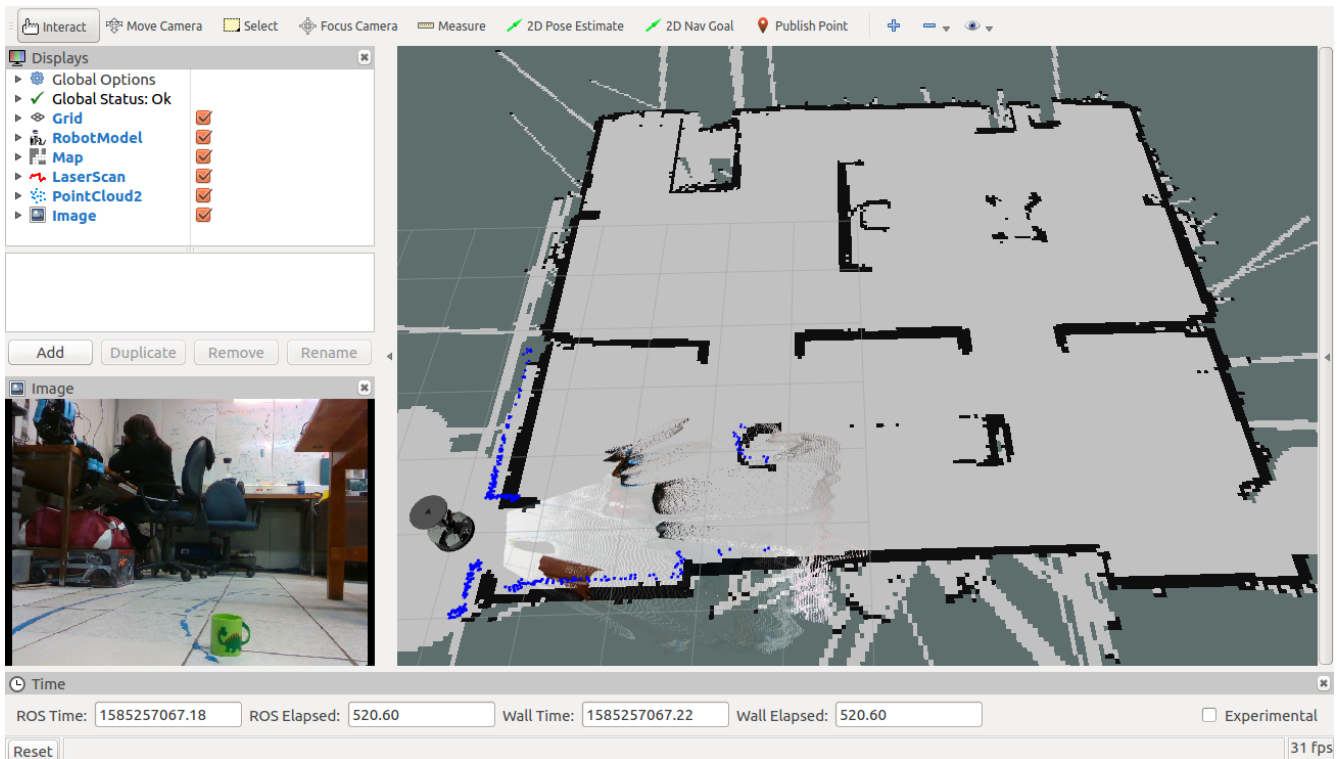
Uso de archivos urdf y árbol de transformaciones con el paquete tf

Primero nos vamos a la consola de Linux, y antes de iniciar cualquier operación o una nueva pestaña en la consola, teclear el comando

```
source MobileRobots-2020-2-for-Covid19/catkin_ws/devel/setup.bash
```

Una vez ejecutado, vamos ahora a introducir el siguiente comando, que nos permite ejecutar el simulador RVIZ

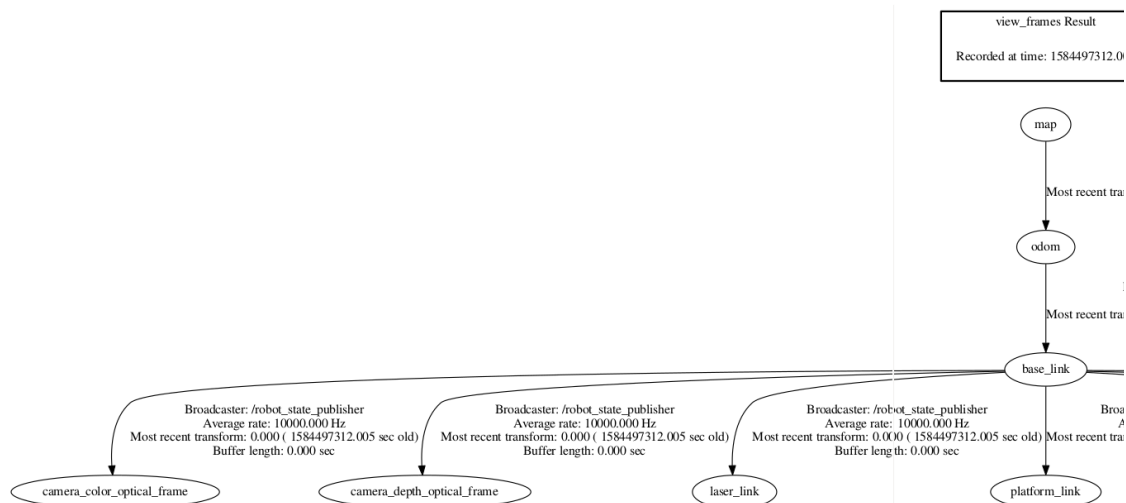
```
roslaunch bring_up robotino simul.launch
```



Posteriormente con el simulador abierto, abrimos otra ventana en la consola, e introducimos el siguiente comando

```
roslaunch tf view_frames
```

Y en la carpeta personal se obtuvo el archivo `frames.pdf`



312.005

Broadcaster: /loc_amcl
Average rate: 10.217 Hz
nt transform: 1584497312.009 (-0.004 sec old)
Buffer length: 4.894 sec

Broadcaster: /mobile_base
Average rate: 30.201 Hz
nt transform: 1584497311.989 (0.015 sec old)
Buffer length: 4.967 sec

Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
nt transform: 0.000 (1584497312.005 sec old)
Buffer length: 0.000 sec

Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1584497312.005 sec old)
Buffer length: 0.000 sec

Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1584497312.005 sec old)
Buffer length: 0.000 sec

Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1584497312.005 sec old)
Buffer length: 0.000 sec

wheel0_link

wheel1_link

wheel2_link

Posteriormente en otra pestaña de la consola, se introdujo rostopic info, desplegando los tópicos /cmd_vel, /scan, /camera/depthregistered/points

```
joelon2412@joelon2412-Dell:~$ rostopic info /cmd_vel
Type: geometry_msgs/Twist
```

Publishers: None

Subscribers:

- * /mobile_base (http://joelon2412-Dell:34721/)

```
joelon2412@joelon2412-Dell:~$ rostopic info /scan
Type: sensor_msgs/LaserScan
```

Publishers:

- * /laser_simulator (http://joelon2412-Dell:44232/)

Subscribers:

- * /loc_amcl (http://joelon2412-Dell:43442/)
- * /rviz (http://joelon2412-Dell:43124/)

```
joelon2412@joelon2412-Dell:~$ rostopic info /camera/depth_registered/points
Type: sensor_msgs/PointCloud2
```

Publishers:

- * /rgb_simul (http://joelon2412-Dell:40009/)

Subscribers:

- * /rviz (http://joelon2412-Dell:43124/)

Posteriormente se detuvo la ejecución del simulador RVIZ, y se modificó el archivo robotino_simul.launch, ubicado en la carpeta catkin_ws/src/bring_up/launch/ y posteriormente introduciendo el comando en esa carpeta

atom robotino_simul.launch //Si está instalado el editor de texto atom

gedit robotino_simul.launch //El editor de textos por default de Linux

Y posteriormente se cambairón los parametros de la descripción del robot y el mapa del ambiente tal y se muestran comentados en la imagen del código

```
<launch>
  <param name="robot_description" textfile="$(find robot_description)/robotino.urdf" /> <!--
  robotino.urdf, justina_simple.urdf -->
  <node pkg="robot_state_publisher" type="robot_state_publisher" name="robot_state_publisher"/>

  <node name="rviz" pkg="rviz" type="rviz" args="-d $(find
  config_files)/rviz/getting_started.rviz"/>
  <node name="laser_simulator" pkg="laser_simulator" type="laser_simulator" output="screen">
    <param name="noise" value="0.05"/>
  </node>
  <node name="mobile_base" pkg="mobile_base_simulator" type="mobile_base_simulator.py"
  output="screen"/>
  <node name="rgbd_simul" pkg="rgbd_simul" type="rgbd_simul2.py" output="screen">
    <param name="bag_file" value="$(find vision)/bags/test.bag"/>
  </node>

  <node name="map_server" pkg="map_server" type="map_server" output="screen"
  args="$(find config_files)/occupancy_grids/Universum.yaml" /> <!-- Universum.yaml,
  Biorobotica.yaml, TMR 2019.yaml -->
  <node name="loc_amcl" pkg="amcl" type="amcl" output="screen" args="scan:=/scan">
    <param name="odom_alpha1" value="0.6"/>
    <param name="odom_alpha2" value="0.6"/>
```

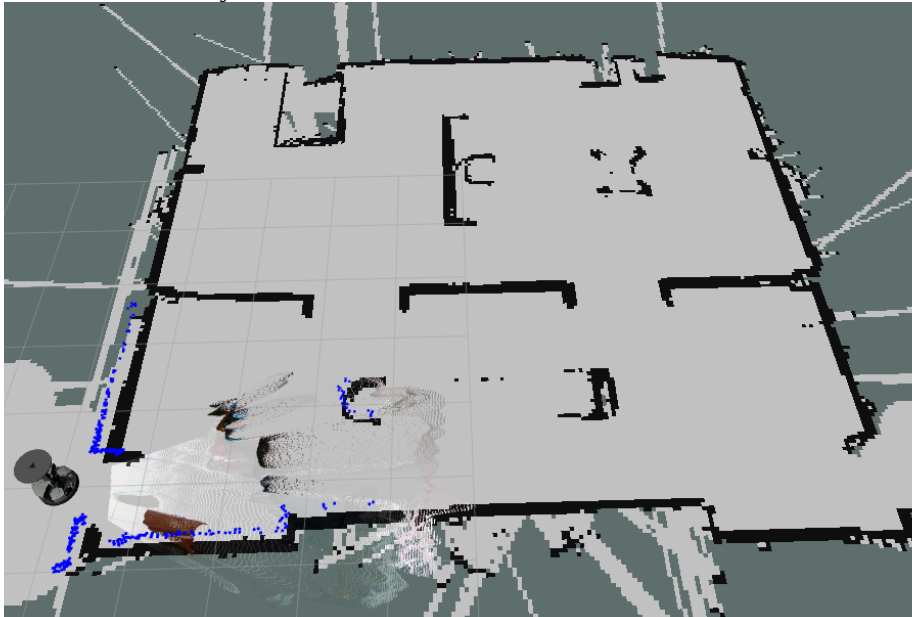
Ahora se van a ver los cambios en el simulador, primero en la pestaña de la carpeta del archivo robotino_simul.launch, cambiandonos a catkin_ws y luego introduciendo el comando catkin_make

Y en la otra pestaña del simulador ejecutar
roslaunch bring_up robotino simul.launch

Haciendo estas acciones por cada cambio que hagamos

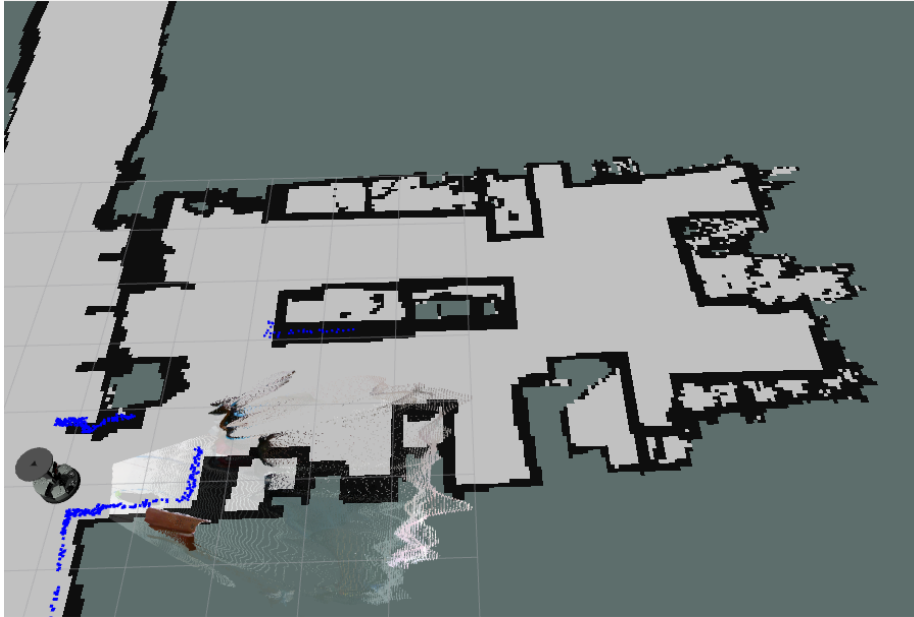
Descripción del robot: robotino.urdf

Mapa del robot: Universum.yaml



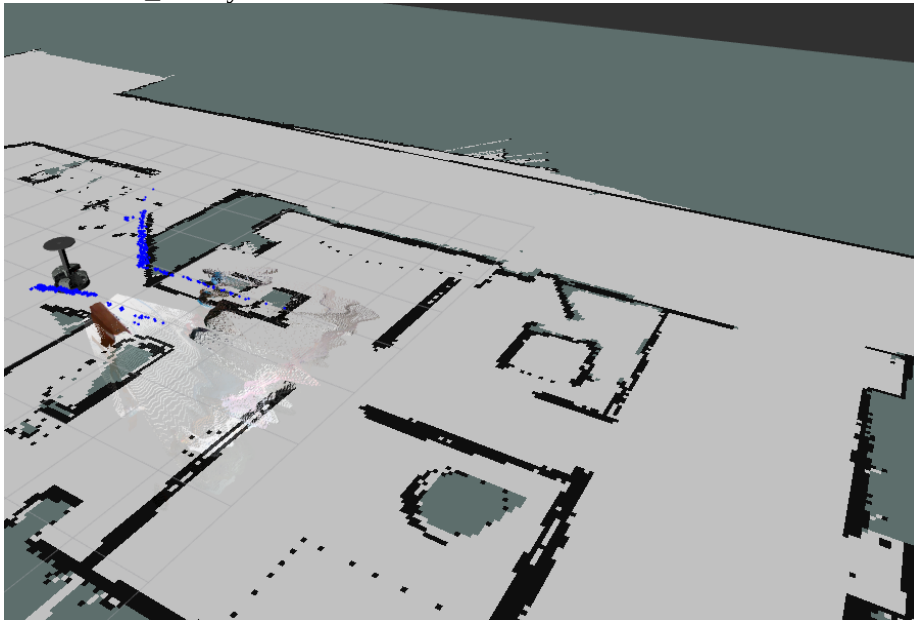
Este es el mapa por default configurado en el programa.

Descripción del robot: robotino.urdf
Mapa del robot: Biorobotica.yaml



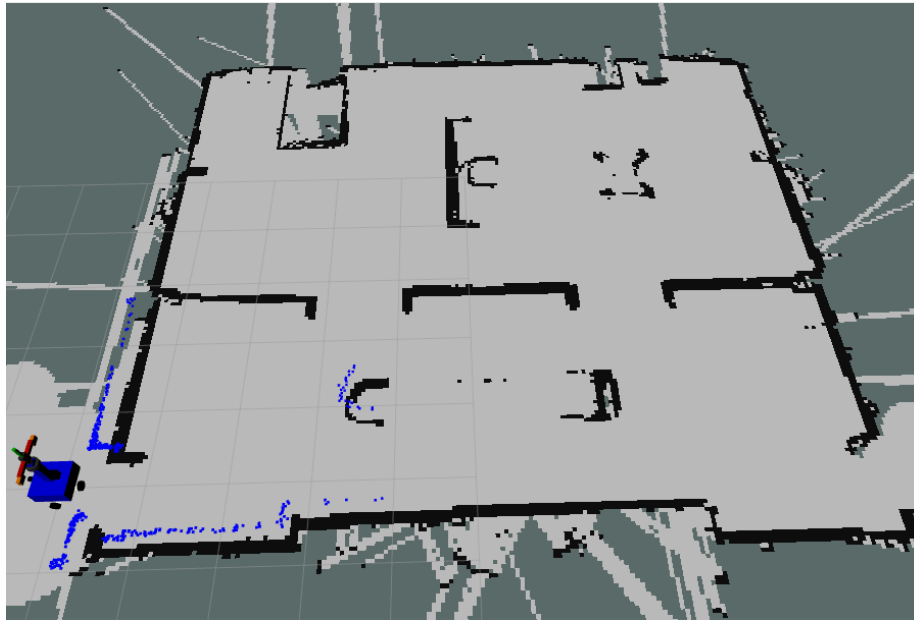
Lo que vemos de este cambio es en la forma del mapa, ya que el anterior era casi plano y con líneas interiores en sí, y este no es un cuadrado como el anterior, en este tiene formas “huecas” y algo encimadas en sí, aunque un poco pixelado.

Descripción del robot: robotino.urdf
Mapa del robot: TMR_2019.yaml

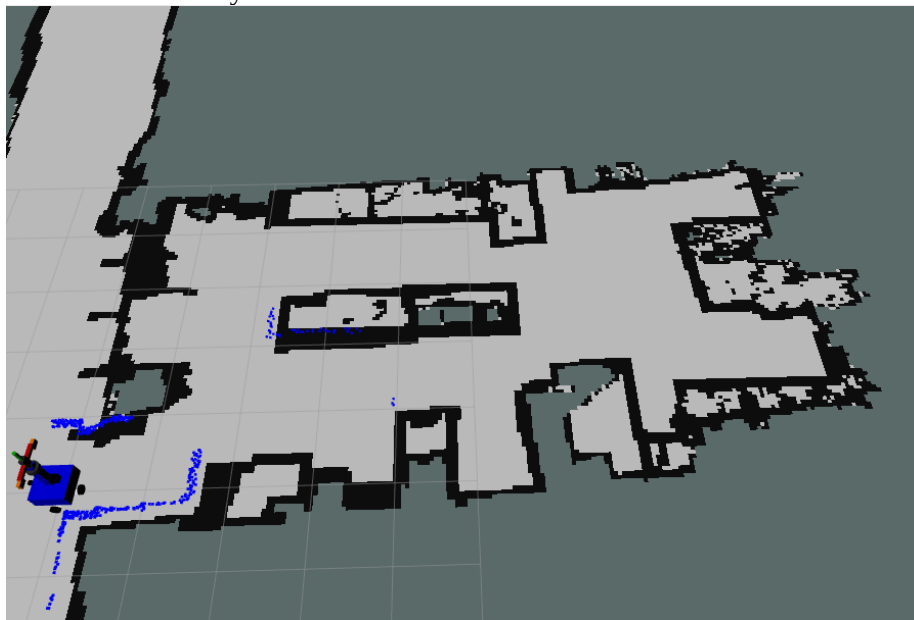


En este tercer y último mapa, es diferente a los otros dos, en este caso se puede observar que es más grande que los otros dos, pero tiene o se quiere asemejar al primer mapa, pero es más extenso y tiene secciones que rodean a ese cuadro, y tiene unos espacios huecos.

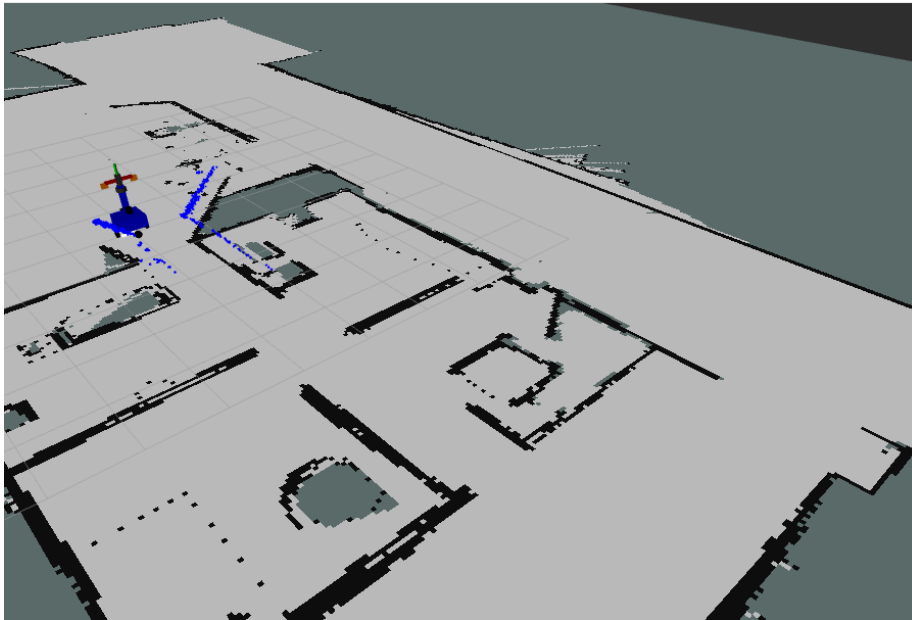
Descripción del robot: justina_simple.urdf
Mapa del robot: Universum.yaml



Descripción del robot: justina_simple.urdf
 Mapa del robot: Biorobotica.yaml



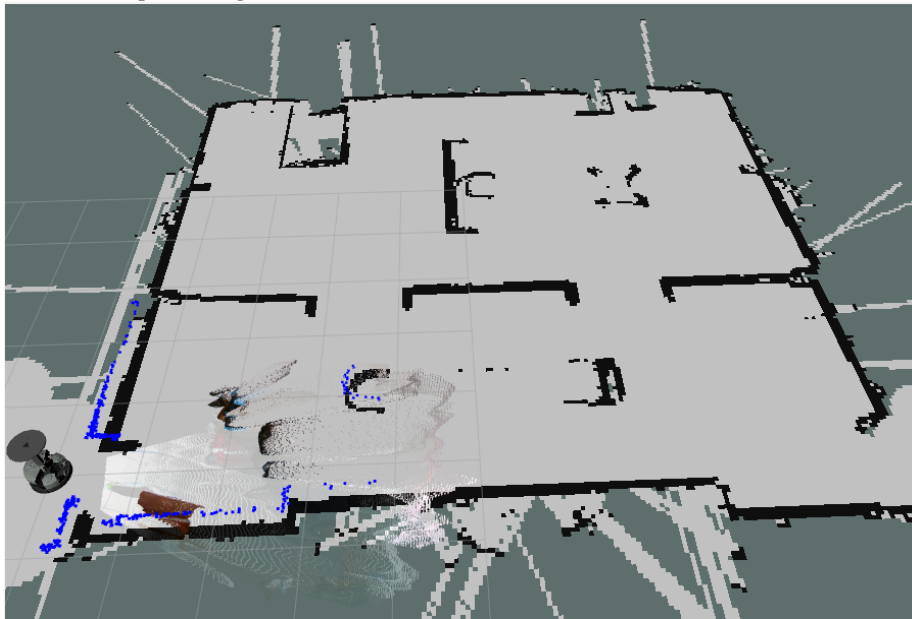
Descripción del robot: justina_simple.urdf
 Mapa del robot: TMR_2019.yaml



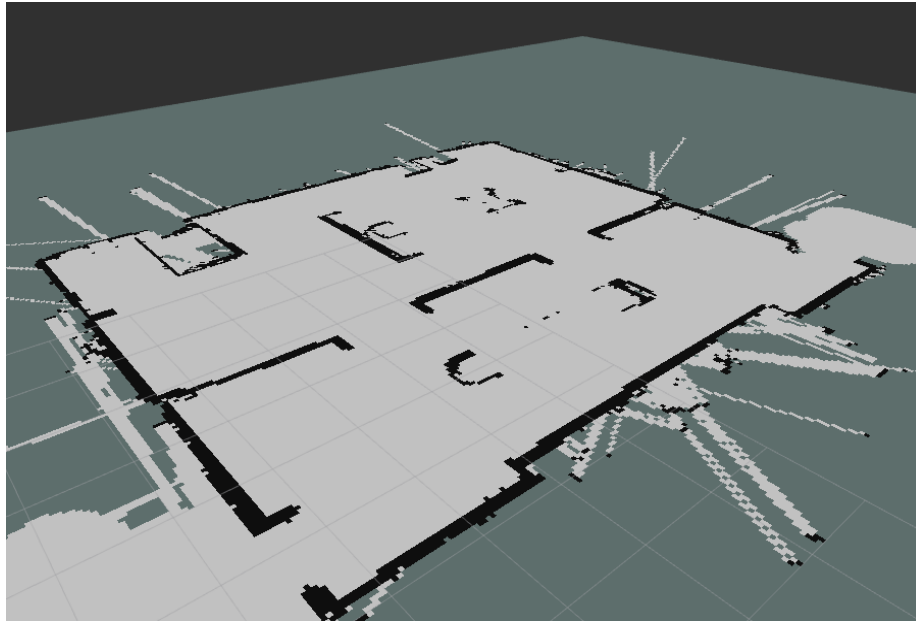
En las últimas tres imágenes, se pudo observar el robot no era gris y con ambas plataformas circulares, en este caso el robot es de color azul en su base y soporte con terminación de punta, pero la “plataforma” superior es una especie como de cuerno color anaranjado.

Posteriormente modificamos el archivo `robotino.urdf` ubicado en la carpeta `catkin_ws/src/hardware/robot_description/`. En este modificamos:

- Los valores de la etiqueta `origin` de la línea 114



- Y eliminando uno de los campos de `<joint>`



Con respecto al cambio de la etiqueta de la línea 114 `<origin>`, no se vió un cambio, permanecio el robot así tal cual como en la primera imagen. Con respecto al cambio de la etiqueta `<joint>` se cambio el “plataform joint”, lo que se observa es que el robot desaparece de la interfaz, pero, si nosotros cambiamos a otro joint como “wheel 0”, “wheel 1” o “wheel 2” nos marca error el programa desde la terminal y no corre el simulador.

Conclusiones

Observamos que si cambiamos diversos programas como `robotino_simul.launch` y `robotino.urdf`, se pueden cambiar los mapas, el diseño del robot, la posición inicial (aunque no se haya notado) y la eliminación de un parametro que “desaparecer” el robot o un elemento (el cual nos marcaba error).

Aprendímos a ejecutar ciertos nodos que visualiza el diseño grafico de la interfaz del robot que deseamos controlar y al mismo tiempo el robot mismo.