

Proyecto Final: Teleoperación y Visualización del Jackal en RViz 2

Instrucciones a seguir para el proyecto final.

Compilacion del workspace ProyectoFinalWS

1. Copia el contenido del directorio **ProyectoFinalWS** en el direction fuente (**src**) de tu **ros2 workspace**.
2. Tu directorio tendra la siguiente estructura.

```
~/ros_ws/src/ProyectoFinalWS$ tree -L 2
.
├── jackal_description
│   ├── CMakeLists.txt
│   ├── launch
│   ├── meshes
│   ├── meta-information.json
│   ├── package.xml
│   ├── rviz
│   └── urdf
├── jackal_mover
│   ├── jackal_mover
│   ├── package.xml
│   ├── resource
│   ├── setup.cfg
│   ├── setup.py
│   └── test
└── ProyectoFinalDescripcion.md
```

3. Configurar el entorno ejecutando:

```
~/ros_ws$ source /opt/ros/jazzy/setup.bash
```

4. Compila los nuevos paquetes utilizando el siguiente comando:

```
~/ros_ws$ colcon build --symlink-install --base-path
src/ProyectoFinalWS/
```

Recuerda hacerlo desde la raiz del workspace.

5. Actualiza el entorno ejecutando:

```
source ~/ros_ws/install/setup.bash
```

Launch file

6. Necesitamos añadir el nodo `teleop_mover` del paquete `jackal_mover` al archivo de lanzamiento.

Puedes hacerlo en el archivo de lanzamiento en Python:

```
~/ros_ws/src/ProyectoFinalWS/jackal_description/launch/display.launch.py
```

o en el archivo en formato XML:

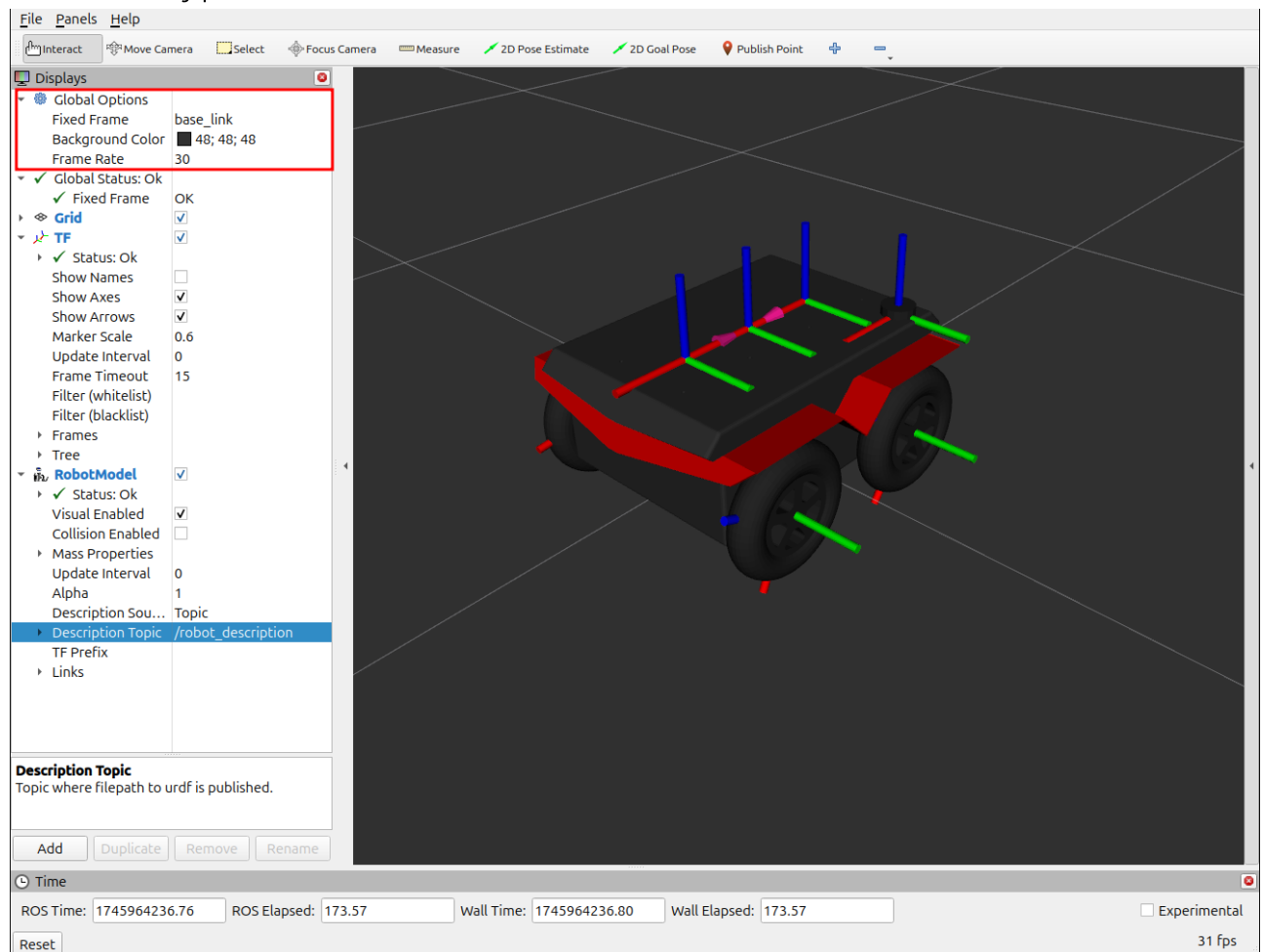
```
~/ros_ws/src/ProyectoFinalWS/jackal_description/launch/display.launch.xml
```

Visualiza tu proyecto y modifica su color

7. Luego de configurar el entorno en una terminal (ver pasos 3 y 5), lanza tu proyecto ejecutando el siguiente comando:

```
ros2 launch jackal_description display.launch.py
```

RViz se abrirá y podrás visualizar el robot Jackal.



8. El color del robot debe coincidir con el color mostrado por el fabricante en su página web:
<https://clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/>.

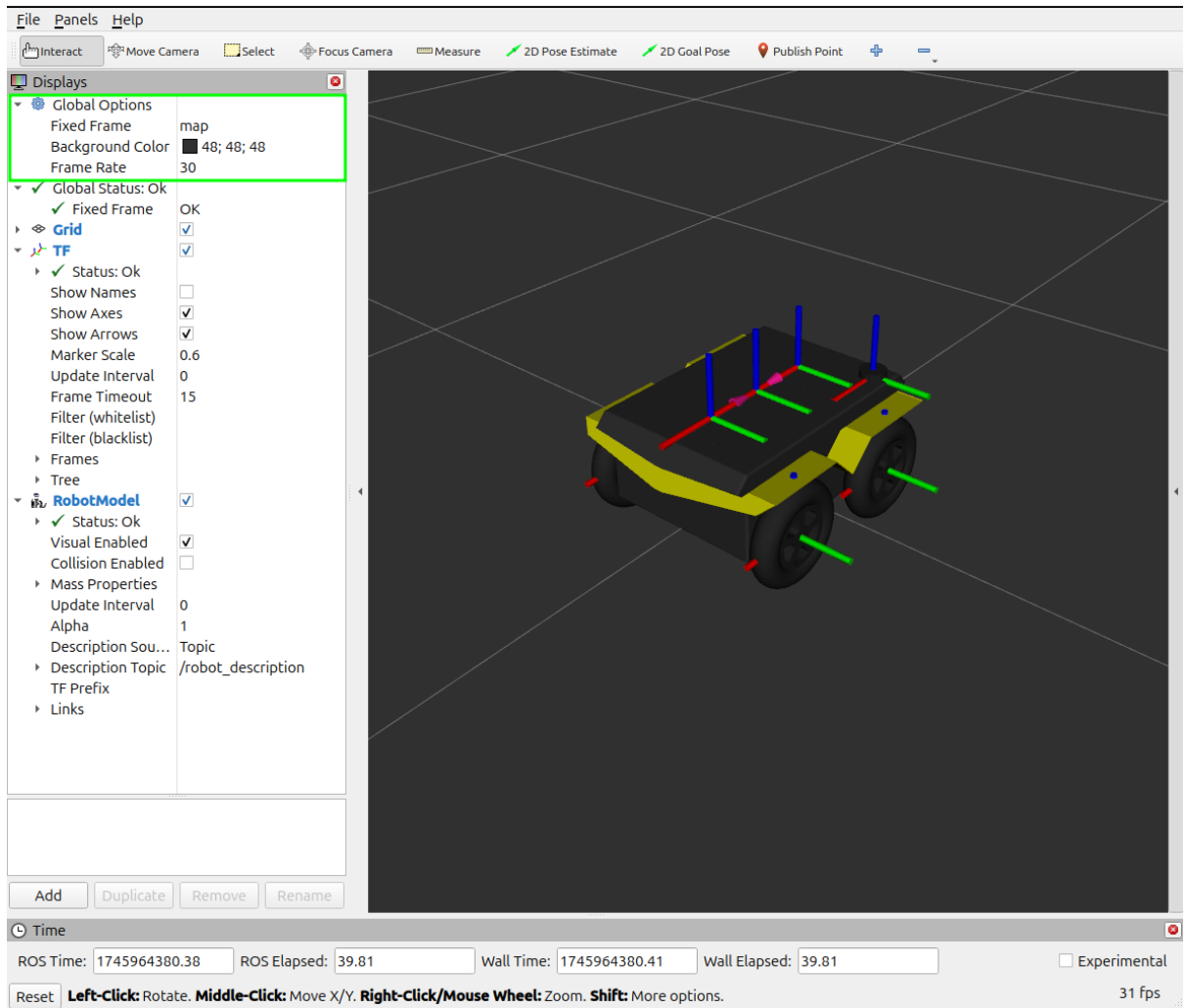
Modifica el color en el archivo:

```
ProyectoFinalWS/jackal_description/urdf/jackal.urdf
```

para que se asemeje al modelo oficial del fabricante.

1. Para visualizar los cambios en el URDF, es necesario detener el comando del paso 7 con **Ctrl + C** y volver a ejecutarlo.

2. Consejo: Revisa los colores definidos en la sección de materiales (<material>) del robot.



Mueve el robot

9. En una nueva terminal, instala el paquete `teleop_twist_keyboard` con el siguiente comando:

```
sudo apt-get install ros-jazzy-teleop-twist-keyboard
```

10. Luego de configurar el entorno en una terminal (ver pasos 3 y 5), ejecuta el controlador con:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

1. Si sigues las instrucciones del nodo `teleop_twist_keyboard`, podrás controlar el movimiento de tu robot usando el teclado.
2. Recuerda seleccionar la terminal donde se está ejecutando `teleop_twist_keyboard` para que los comandos tengan efecto.

Entregable

The screenshot displays the ROS2 GUI interface. On the left, there are panels for 'Nodes' (listing 'Add python and more'), 'Services' (with 'Stop', 'Run', and 'Buffer' buttons), 'Publishers', 'Subscribers', and 'Timers'. The main window is divided into two sections. The top section shows a 2D plot of 'XY Pos' with a blue dot at the origin (0,0) and a red line representing the robot's path. The bottom section shows a 3D view of a robot model with a yellow body and blue wheels, positioned on a grid. The bottom status bar indicates 'ROS Time: 15.08', 'ROS Elapsed: 1908.22', 'Wall Time: 574596355.11', 'Wall Elapsed: 1908.22', and 'Experimental' mode is enabled.

- La terminal donde se está ejecutando `teleop_twist_keyboard`.
- RViz mostrando el robot en movimiento.