

# Robot operációs rendszerek és fejlesztői ökoszisztémák

## Vizualizáció, szimuláció

---

Gincsiné Szádeczky-Kardoss Emese

2025. október 13.

**KUKA**



**iit**

# Tartalom

---

Visszatekintés

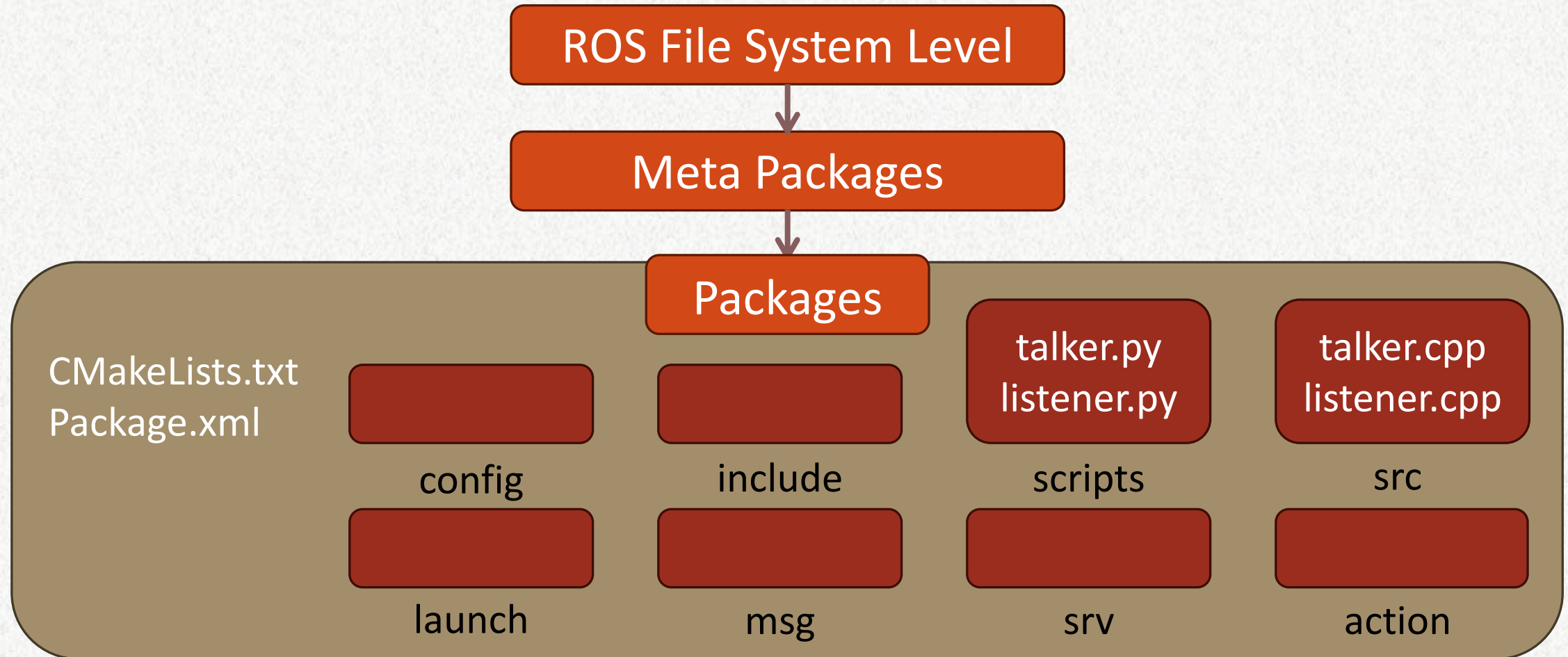
Vizualizáció

Szimuláció

# Visszatekintés

---

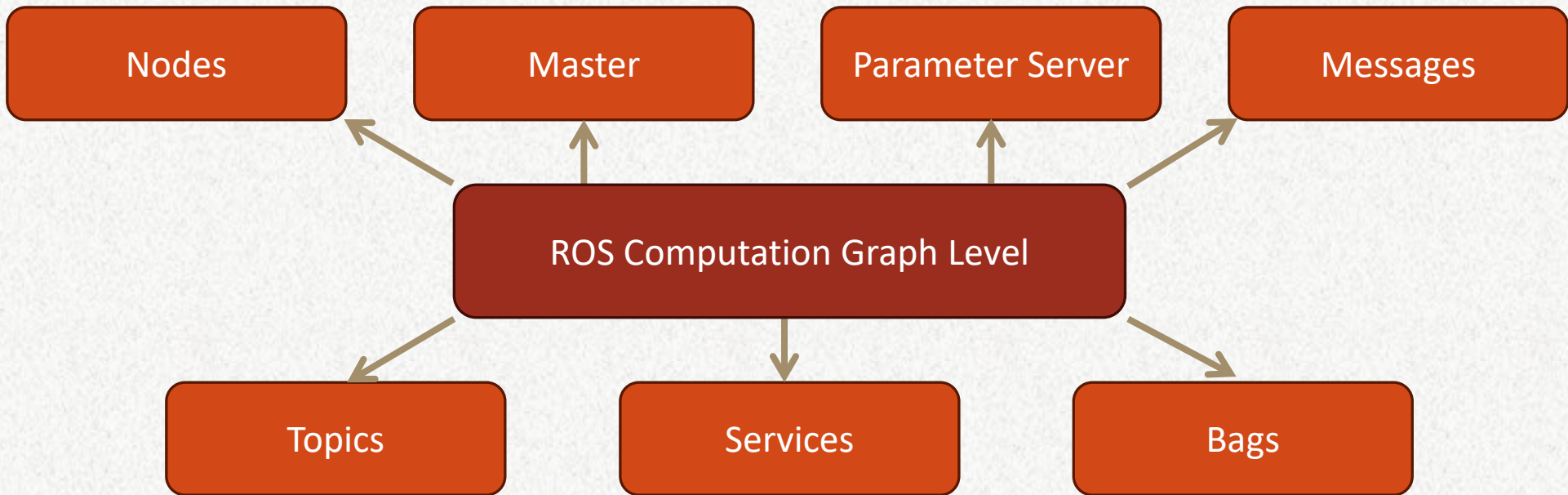
# ROS bevezető





# ROS bevezető

---



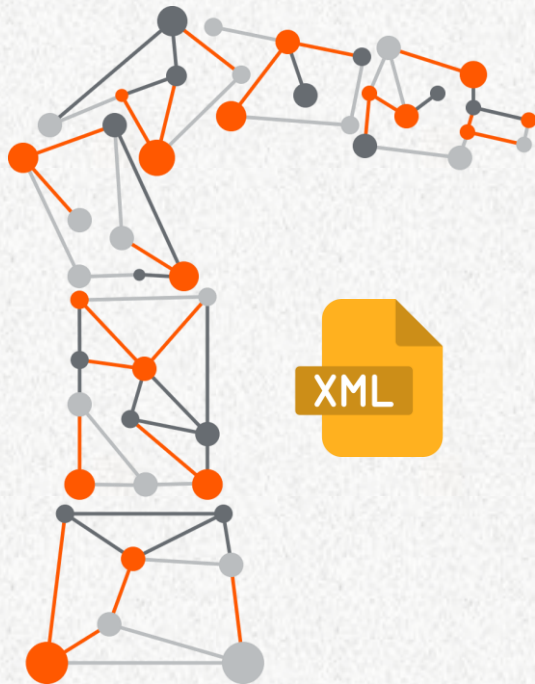
# ROS bevezető – Parancssori utasítások

---

- `catkin_create_pkg`, `catkin_make`: új package létrehozása és buildelése
- `rospack`, `rostack`, `roscdep`: eszköz a package-ek, stack-ek, függőségek kezeléséhez
- `roscd`, `rosls`: package könyvtár váltás és ROS package tartalmának listázása
- `roscp`, `roscd`: fájl másolás egy package-ből, fájl szerkesztése
- `roscd`, `rosls`: package futtatható állományának/állományainak futtatása
- `roscd`: roscore futtatása (ROS rendszermag: master + roscd + parameter server)
- `roscd`: eszköz node-okhoz
- `roscd`, `rostopic`, `rossrv`, `rosservice`: eszköz node-ok kommunikációjához
- `roscd`: paraméterek kezelése
- `roscd`: adatok logolása, betöltése

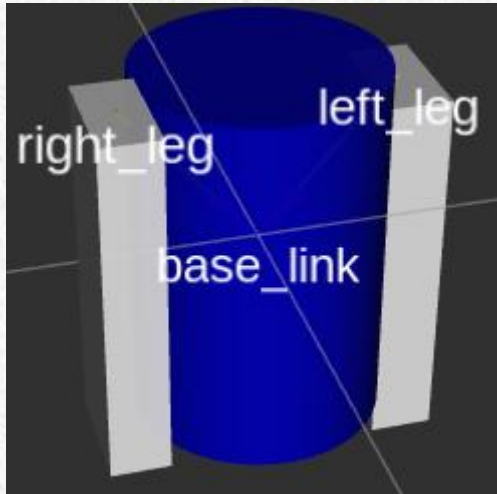
# Unified Robotics Description Format

---



- XML-alapú
- Összekapcsolt merev testek (multibody) modellezése
- Nem része a ROS-nak (más robotikai alkalmazások is használják)
- URDF elemei:
  - `<link>`
  - `<joint>`

# URDF - Példa



```
1 <?xml version="1.0"?>
2 <robot name="materials">
3
4   <material name="blue">
5     <color rgba="0 0 0.8 1"/>
6   </material>
7
8   <material name="white">
9     <color rgba="1 1 1 1"/>
10  </material>
11
12  <link name="base_link">
13    <visual>
14      <geometry>
15        <cylinder length="0.6" radius="0.2"/>
16      </geometry>
17      <material name="blue"/>
18    </visual>
19  </link>
20
21
```

```
22 <link name="right_leg">
23   <visual>
24     <geometry>
25       <box size="0.6 0.1 0.2"/>
26     </geometry>
27     <origin rpy="0 1.57075 0" xyz="0 0 -0.3"/>
28     <material name="white"/>
29   </visual>
30 </link>
31
32 <joint name="base_to_right_leg" type="fixed">
33   <parent link="base_link"/>
34   <child link="right_leg"/>
35   <origin xyz="0 -0.22 0.25"/>
36 </joint>
37
38 <link name="left_leg">
39   <visual>
40     <geometry>
41       <box size="0.6 0.1 0.2"/>
42     </geometry>
43     <origin rpy="0 1.57075 0" xyz="0 0 -0.3"/>
44     <material name="white"/>
45   </visual>
46 </link>
47
48 <joint name="base_to_left_leg" type="fixed">
49   <parent link="base_link"/>
50   <child link="left_leg"/>
51   <origin xyz="0 0.22 0.25"/>
52 </joint>
53
54 </robot>
```

urdf\_tutorial/urdf/04-materials.urdf



# Vizualizáció

---

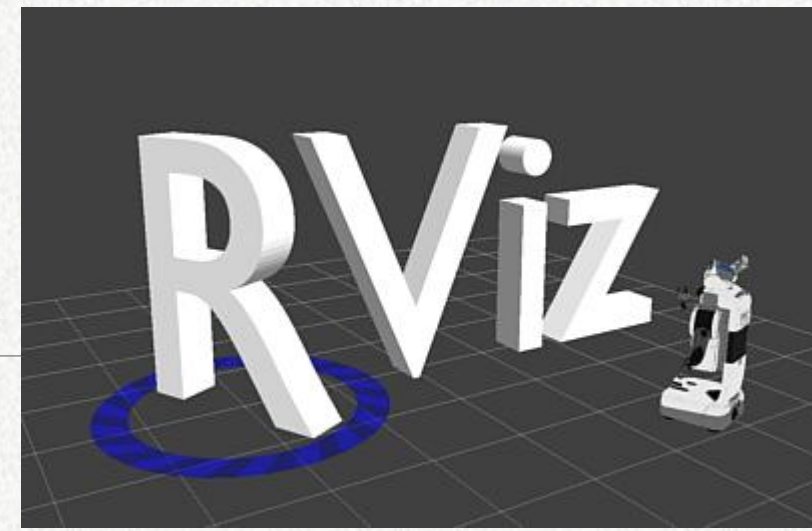
# ROS megjelenítési eszközök

---

- Vizualizáció
  - [rqt\\_gui](#) – ROS integrált GUI
  - [rqt\\_image\\_view](#) – GUI képek megjelenítéséhez
  - [rviz](#) – ROS 3D vizualizációs eszköze
- Plotting
  - [rqt\\_plot](#) – numerikus értékek 2D megjelenítése
- Debug
  - [tf\\_echo](#), [view\\_frames](#), [rqt\\_graph](#)
- Loggolás
  - [rqt\\_console](#) – GUI konzol tartalmának loggolásához
  - [rqt\\_bag](#) – GUI ROS bag fájlok megjelenítéséhez, visszajátszásához

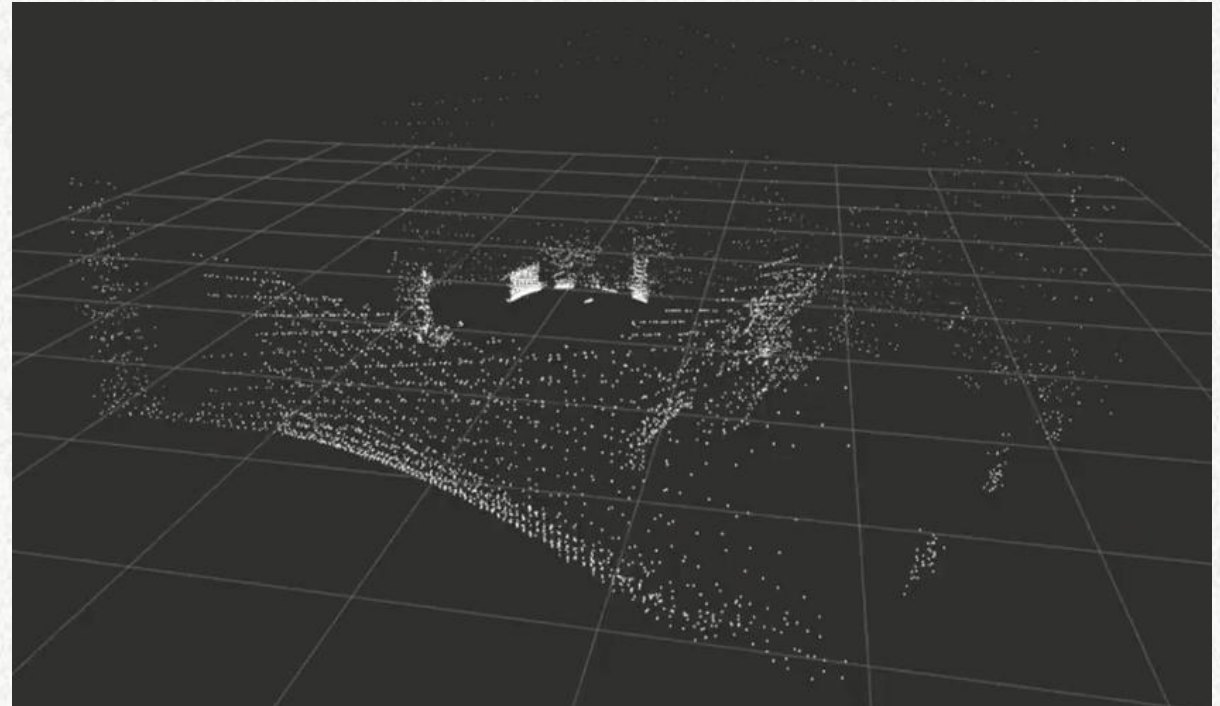
# rviz

- <http://wiki.ros.org/rviz>
- ROS 3D vizualizációs környezete
- Robot modell, keretek, szenzor információ és egyéb adatok megjelenítése
- Ábrázolható, hogy mit lát a robot, mit gondol a robot, mit csinál a robot
- Saját (szoftveresen generált) marker-ek is megjeleníthetők
- Kombinált nézet generálás
- Debuggolást, validálást könnyíti
- Free, open source



# Néhány robotikai érzékelő

- Belső érzékelők – saját állapot mérésére:
  - enkóderek
  - IMU – inertial measurement unit
- Bumper
- Gépi látás
  - Mono vagy sztereo kamera
  - RGBD – mélységkép
- Távolságmérők
  - ultrahangos, lézeres
- LiDAR
  - pontfelhők

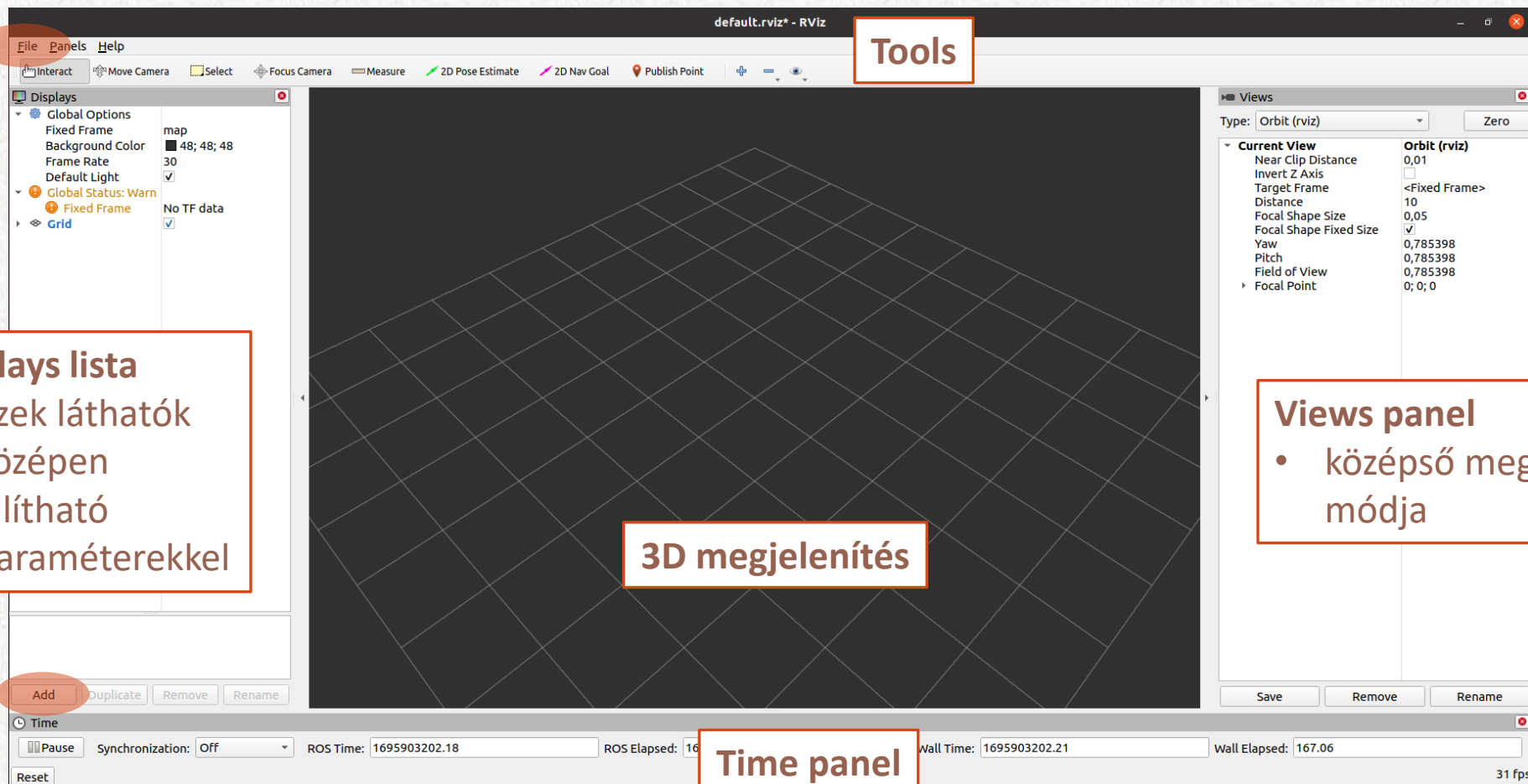




# rviz

Konfiguráció  
mentése,  
betöltése  
(\* .rviz fájlok)

```
emese@ubuntu:~$ rviz
```



Új display  
hozzáadása

3D megjelenítés

Tools

## Displays lista

- ezek láthatók középen
- állítható paraméterekkel

## Views panel

- középső megjelenítés módja

## Time panel

# tf

---

- Részletes leírás: <http://wiki.ros.org/tf> illetve <http://wiki.ros.org/tf2>
- Transzformációs könyvtár
- 3D keretek (koordináta-rendszerek, *frames*) megjelenítésére, vizsgálatára szolgál
- Időbeli változásokat is tud kezelni
- Különböző objektumok pozícióját, orientációját lehet vele meghatározni bármelyik keretben.
- A robotunk rendszerint sok keretet tartalmaz (ld. urdf)
- Keretek fa struktúrájú hierarchiáját kezeli
- Koordinátatranszformációkat támogatja

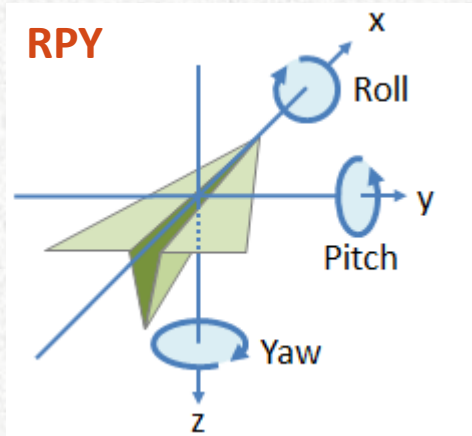
# tf

---

Két alapvető tf-hez kapcsolódó feladat van:

- Megosztani a többiekkel frame-ek helyzetét (broadcasting transforms)
  - [tf2 broadcaster C++](#)
  - [tf2 broadcaster Python](#)
- Másoktól információt kapni frame-ekről (listening for transforms)
  - [tf2 listener C++](#)
  - [tf2 listener Python](#)
- Konzol eszközök is rendelkezésre állnak
  - [tf\\_echo](#), [tf\\_monitor](#), [view\\_frames](#), [roswtf](#), [static transform publisher](#)

# Szöghelyzet jellemzése: RPY, kvaterniók

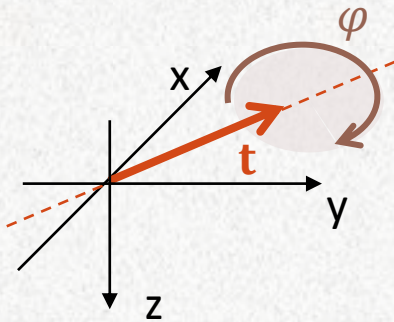


A **kvaterniók** halmaza ( $\mathbb{Q}$ ) a háromdimenziós tér ( $\mathbb{R}^3$ ) kiterjesztése:

$$\mathbf{q} = (s, \mathbf{w}), \mathbf{q} \in \mathbb{Q}, s \in \mathbb{R}, \mathbf{w} = [x, y, z]^T \in \mathbb{R}^3, \mathbf{q} = s + xi + yj + zk$$

Egy  $\mathbf{t}$  irányvektorral adott tengely körüli  $\varphi$  szögű elfordulás ( $\|\mathbf{t}\| = 1$ ) felírható kvaternióval:

$$\mathbf{q} = \left( \cos \frac{\varphi}{2}, \sin \frac{\varphi}{2} \mathbf{t} \right)$$

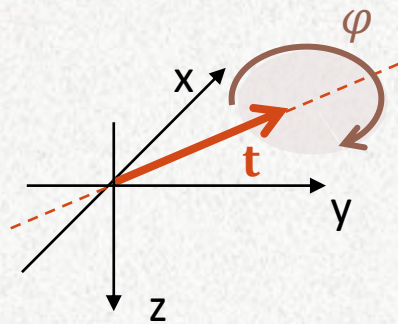
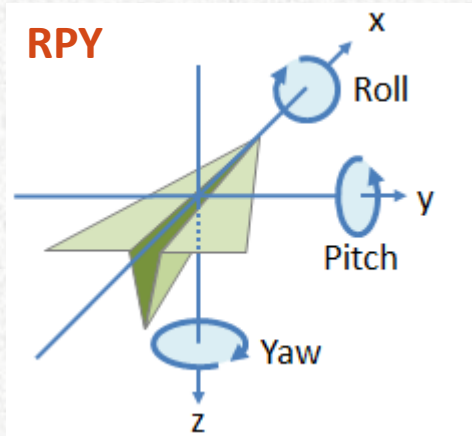


Ha az orientáció kvaternióval adott, meghatározható belőle a  $\mathbf{t}$  tengely és a  $\varphi$  szög:

$$\varphi = 2 \arccos s, \mathbf{t} = \mathbf{w} / \sin \frac{\varphi}{2}$$



# Szöghelyzet jellemzése: RPY, kvaterniók



A **kvaterniók** halmaza ( $\mathbb{Q}$ ) a háromdimenziós tér ( $\mathbb{R}^3$ ) kiterjesztése:

$$\mathbf{q} = (s, \mathbf{w}), \mathbf{q} \in \mathbb{Q}, s \in \mathbb{R}, \mathbf{w} = [x, y, z]^T \in \mathbb{R}^3, \mathbf{q} = s + xi + yj + zk$$

Eg fel

```
emese@ubuntu:~$ rosmmsg info geometry_msgs/Pose )
geometry_msgs/Point position
float64 x
float64 y
float64 z
geometry_msgs/Quaternion orientation
float64 x
float64 y
float64 z
float64 w
```

Ha ter

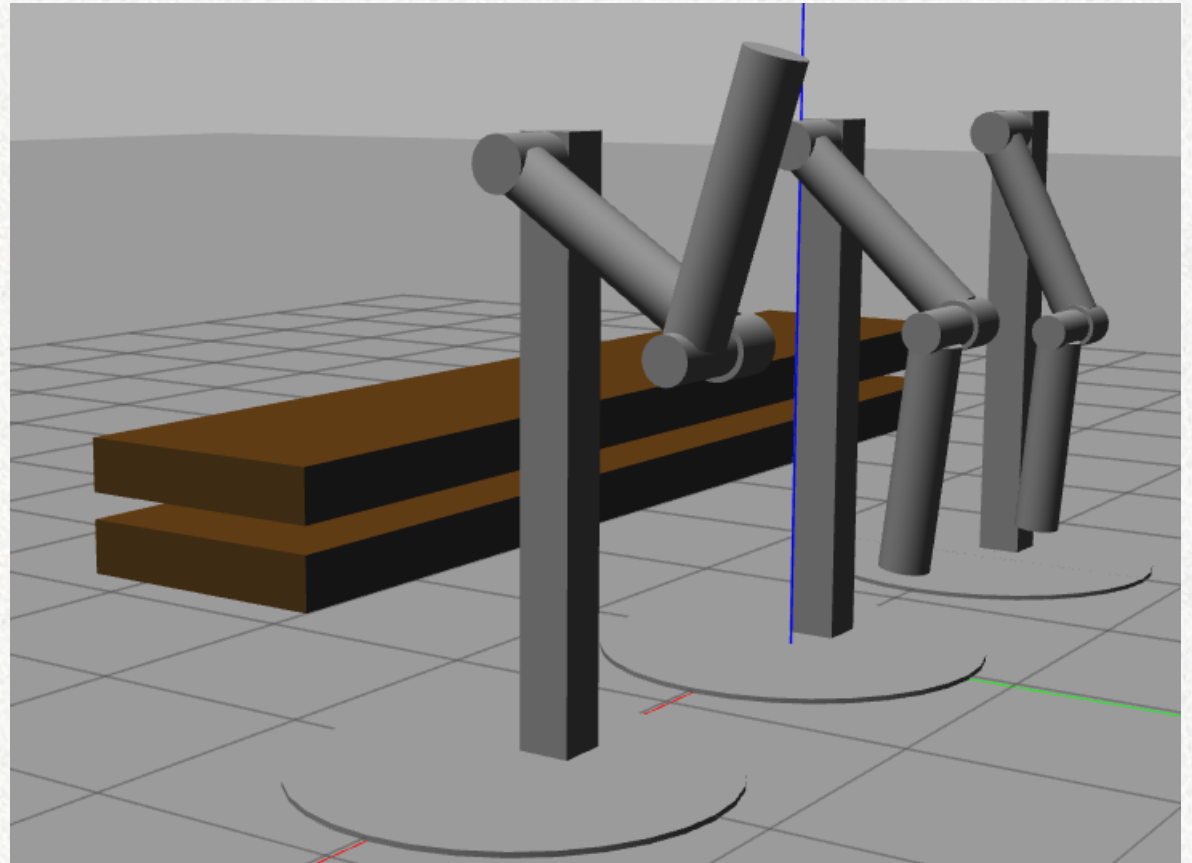
$$\varphi = 2 \arccos s, \mathbf{t} = \mathbf{w} / \sin \frac{\varphi}{2}$$

# Szimuláció

---

# Mi a robotszimuláció?

- Beágyazott robot alkalmazás fejlesztése
- A robot fizikai jelenléte, használata nélkül
- Robotok tervezéséhez, algoritmusok teszteléséhez, MI alapú megoldások tanításához érdemes használni



# Szimuláció előnyei

---

- Költség szempontjából
  - Nincs szükség a berendezésre
  - Nincs sérülés, károsodás veszély
  - A hardver nem igényel erőforrást és nem használódik el
  - Nincs szükség az eszköz karbantartására
  - Humán erőforrás nincs veszélyben
- Idő szempontjából
  - Szimulációk futtathatók párhuzamosan
  - Nem kell akkumulátorokat újratölteni
- Tapasztalatok szempontjából
  - Tetszőleges környezetben, tetszőleges robotok, tetszőleges szenzorok szimulálhatók
  - Megismételhetők a kísérletek
  - Skálázhatóak a szimulációk





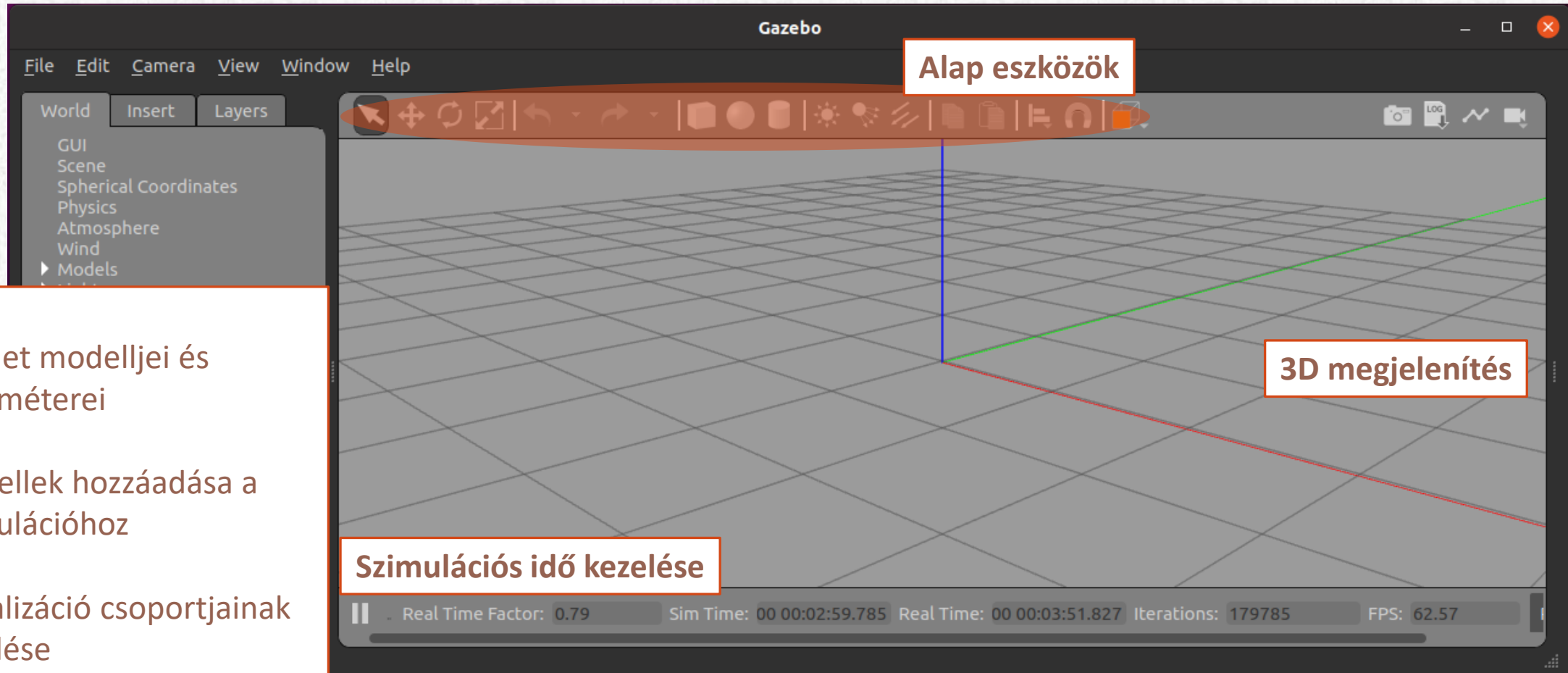
# Gazebo

---

- <http://classic.gazebo.org/>
- 3D robot szimulátor
- Több robot egyidejű szimulációja is végezhető
- Dinamikai szimulációk különböző solverekkel
- Realisztikus 3D megjelenítés (megvilágítás, árnyékok, textúrák)
- Szenzorok és mérési pontatlanságok szimulálása
- Sok beépített robot modell, de saját is építhető
- gzserver – fizikai szimuláció számításai (pl. szenzor adatok generálása)
- gzclient – grafikus felhasználói felületet biztosítja
- Cloud simulation lehetséges (pl. gzserver futtatása)

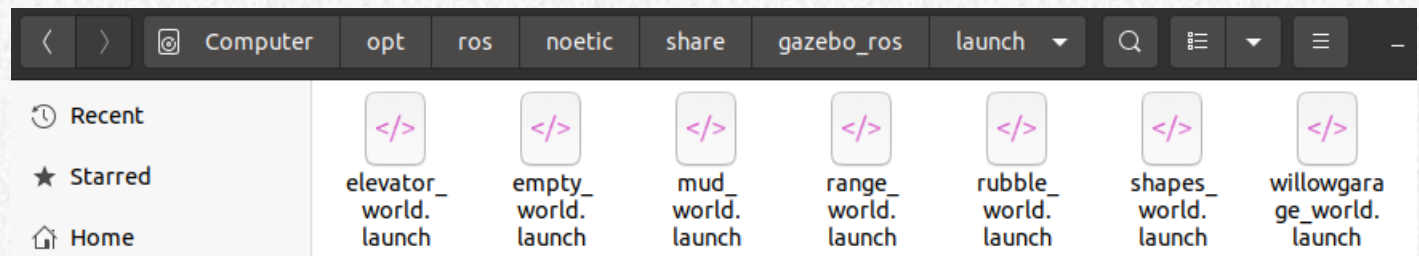
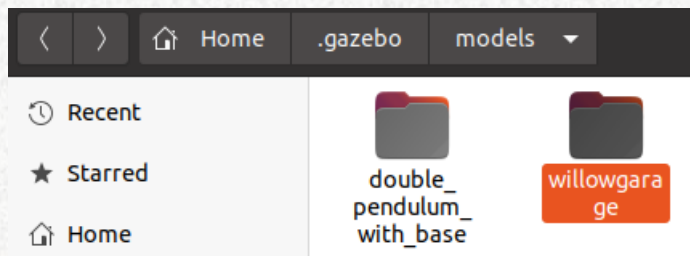
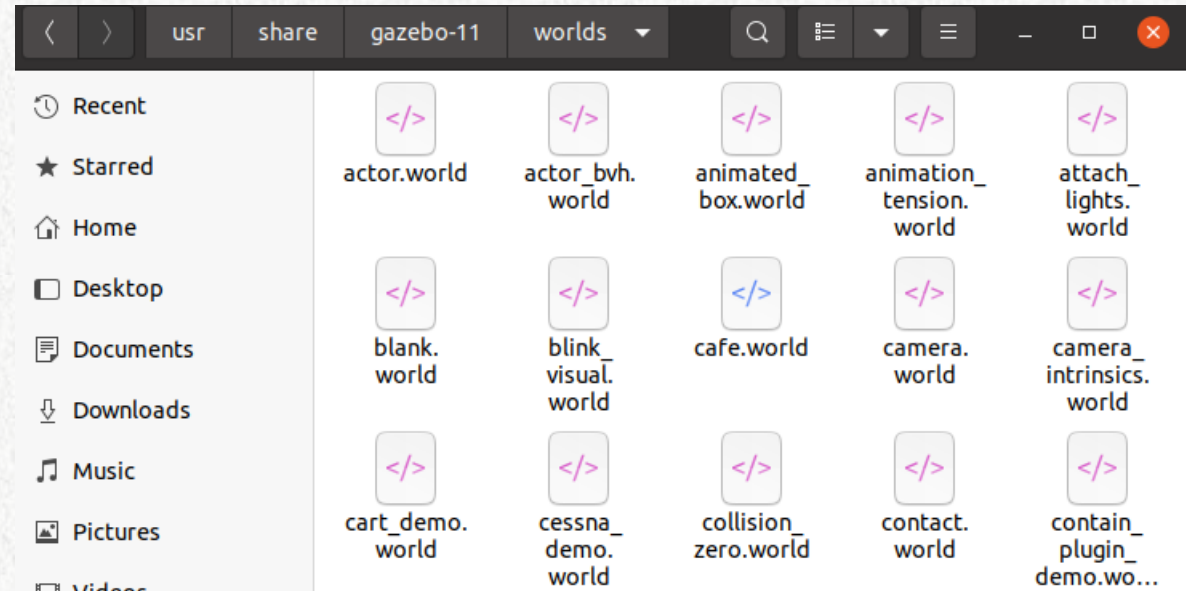
# Gazebo - felület

```
emese@ubuntu:~$ gazebo
```



# Gazebo

- Gazebo worlds (\*.world)
- Gazebo models (\*.sdf)
- Launch files (\*.launch)



# Gazebo

- Simulation Description Format: [SDF](#)
  - Gazebo modellek és world fájlok formátuma
  - XML formátum, ami a robot környezetét és az objektumokat írja le szimulációhoz, vizualizációhoz
  - Robotok, statikus és dinamikus objektumok, megvilágítás, terep és fizikai tulajdonságok megadása
  - URDF egy robot leírását tudja, míg most akár több modellt és a környező világot és kölcsönhatásokat is kell modellezni.
- [URDF és Gazebo](#)

```
Open [icon] empty.world [Read-Only]
~/usr/share/gazebo-11/worlds

1 <?xml version="1.0" ?>
2 <sdf version="1.5">
3   <world name="default">
4     <!-- A global light source -->
5     <include>
6       <uri>model://sun</uri>
7     </include>
8     <!-- A ground plane -->
9     <include>
10      <uri>model://ground_plane</uri>
11    </include>
12  </world>
13 </sdf>
```

```
model.sdf
~/gazebo/models/double_pendulum_with_base

1 <?xml version="1.0" ?>
2 <sdf version="1.5">
3   <model name="double_pendulum_with_base">
4     <link name="base">
5       <inertial>
6         <mass>100</mass>
7       </inertial>
8       <visual name="vis_plate_on_ground">
9         <pose>0 0 0.01 0 0 0</pose>
10        <geometry>
11          <cylinder>
12            <radius>0.8</radius>
13            <length>0.02</length>
14          </cylinder>
15        </geometry>
16        <material>
17          <script>
18            <uri>file://media/materials/scripts/gazebo.material</uri>
19            <name>Gazebo/Grey</name>
20          </script>
21        </material>
22      </visual>
23      <visual name="vis_pole">
24        <pose>-0.275 0 1.1 0 0 0</pose>
25        <geometry>
26          <box>
27            <size>0.2 0.2 2.2</size>
28          </box>
29        </geometry>
30        <material>
31          <script>
32            <uri>file://media/materials/scripts/gazebo.material</uri>
33            <name>Gazebo/Grey</name>
34          </script>
35        </material>
36      </visual>
37      <collision name="col_plate_on_ground">
38        <pose>0 0 0.01 0 0 0</pose>
39        <geometry>
40          <cylinder>
41            <radius>0.8</radius>
42            <length>0.02</length>
43          </cylinder>
44        </geometry>
45      </collision>
```



# További robot szimulátorok

---

- [CoppeliaSim](#) (V-REP)
  - [ROS Tutorials](#)
  - [ROS Interfaces](#)
- [Webots](#) – Open source robot simulator
  - [ROS for Webots](#)
- [MuJoCo](#) – Advanced physics simulation