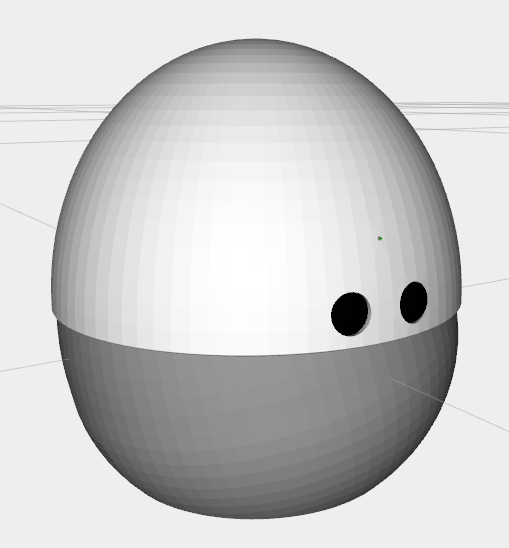
Unit 1: Creating the Visual Robot Model with URDF



In this unit, you will learn how to go from a physical robot to a visual virtual model. By visual, we understand that its not a physically working simulation model. It's only the barebones of what, at the end, will be used for simulation.  
But this URDF Visual model is already very useful.  
If you have a real robot and you want to use the ROS infrastructure, you need a virtual description of how the robot is connected and where each of the sensors is in some applications. For example, if you have a camera mounted on the head of the robot, through the virtual robot description (the URDF file ), you can use the TF ROS structure to know exactly where the camera is based by only the joint sensor readings.  
It also allows you to represent the robot model inside RVIZ.

Warning

Before starting, please check that you have terminated any of the scripts that you launched in the previous unit.

Create your first URDF Model

So, you will start by creating the URDF of the robot Mira.  
For this, you will follow these steps:

* Learn how to use the URDF creation tools and the step-by-step procedure for creating a robot model.
* Learn about the morphology of the robot you want to work with.
* Obtain the 3D models that you will need in the correct format.
* Generate the link and joint structure.
* Test the movement of the joints.

So, let's get started.

1. Learn how to use the URDF creation tools and the creation procedure

Let's create the URDF file in the appropriate ROS structure.  
Let's create a ROS package for each robot model that you create.  
In this case, create a ROS package named "my\_mira\_description." It's a common practice to always create a "my\_robot\_description" package where you store all of the files that describe the robot. You will find this robotname\_description package everywhere in ROS packages that have robot models defined.

|  |
| --- |
| Execute in WebShell #1 |

In [ ]:



cd **/**home**/**user**/**catkin\_ws**/**src

In [ ]:



catkin\_create\_pkg my\_mira\_description rospy rviz controller\_manager gazebo\_ros joint\_state\_publisher robot\_state\_publisher

Once you have the "my\_mira\_description" package, create the following folders inside it:

* launch
* models
* rviz\_config
* config
* urdf
* worlds

|  |
| --- |
| Execute in WebShell #1 |



cd /home/user/catkin\_ws/src/my\_mira\_description

mkdir launch

mkdir models

mkdir rviz\_config

mkdir config

mkdir urdf

mkdir worlds

These folders are the ones that you will need to have a fully-functional simulated robot.  
Now, create a URDF file called "mira.urdf" in the "urdf" folder

|  |
| --- |
| Execute in WebShell #1 |

In [ ]:



cd **/**home**/**user**/**catkin\_ws**/**src;rospack profile

In [ ]:



roscd my\_mira\_description

In [ ]:



touch urdf**/**mira\_simple.urdf

You can also create it through the IDE.

Now, create the simplest URDF:

mira\_simple.urdf

In [ ]:



**<?**xml version**=**"1.0"**?>**

**<**robot name**=**"mira"**>**

**<**link name**=**"base\_link"**>**

**<**visual**>**

**<**origin rpy**=**"0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder radius**=**"0.06" length**=**"0.09"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

**<**link name**=**"roll\_M1\_link"**>**

**<**visual**>**

**<**origin rpy**=**"0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder radius**=**"0.06" length**=**"0.09"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

**<**joint name**=**"roll\_joint" type**=**"revolute"**>**

**<**parent link**=**"base\_link"**/>**

**<**child link**=**"roll\_M1\_link"**/>**

**<**origin xyz**=**"0.0023 0 -0.0005" rpy**=**"0 0 0"**/>**

**<**limit lower**=**"-0.2" upper**=**"0.2" effort**=**"0.1" velocity**=**"0.005"**/>**

**<**axis xyz**=**"1 0 0"**/>**

**</**joint**>**

**</**robot**>**

END mira\_simple.urdf

1. All of the measurements in URDF are in the International Unit System. Therefore, meters for distance, radians for angles, and kilograms for weight.
2. There are three basic geometry shapes that can be used: box, cylinder, and sphere.

In [ ]:



**<**cylinder radius**=**"0.06" length**=**"0.09"**/>**

**<**box size**=**"0.0005 0.0005 0.0005"**/>** x**-**length,y**-**legth **and** z**-**length

**<**sphere radius**=**"0.06"**/>**

2. Links and Joints

In the example given, you have two links; in this case, two cylinders connected through a joint. Joints are what make elements of a robot turn and move. They are the articulations of the robot.

The main elements to define in a joint are:

* Type: there are these types: revolute, continuous, prismatic, fixed, floating, and planar. You can learn more here: http://wiki.ros.org/urdf/XML/joint The joint selection will depend on how the physical model of your robot moves.
* Parent and Child: Here is where you set who is connected to your link.
* Origin: All of the coordinates and rpy are referenced to the Parent axis, not the child axis.
* Limit: This is a very important element, especially when you have to control a robot movement.
* Axis: Here you define around which Parent's AXIS the Child link will revolve. This, of course, depends on the type of joint; some of them don't have axis tags because they are irrelevant, such as the fixed joint.

3. See the URDF

Once you have done this, it is time to see the result. To know how ROS will see the model and to help you position the links and joints, you will use the following launch. Create it in a file called **urdf\_visualize.launch**, inside the *launch* directory of your package.

|  |
| --- |
| Execute in WebShell #1 |

In [ ]:



cd **/**home**/**user**/**catkin\_ws**/**src

In [ ]:



roscd my\_mira\_description

In [ ]:



touch launch**/**urdf\_visualize.launch

In [ ]:



**<**launch**>**

​

**<!--** USE: roslaunch my\_mira\_description urdf\_visualize.launch model:**=**'$(find myrobot\_package)/urdf/myrobot.urdf' **-->**

**<**arg name**=**"model" default**=**""**/>**

​

​

**<**param name**=**"robot\_description" command**=**"cat $(arg model)" **/>**

​

**<!--** send fake joint values **-->**

**<**node name**=**"joint\_state\_publisher" pkg**=**"joint\_state\_publisher" type**=**"joint\_state\_publisher"**>**

**<**param name**=**"use\_gui" value**=**"TRUE"**/>**

**</**node**>**

​

**<!--** Combine joint values **-->**

**<**node name**=**"robot\_state\_publisher" pkg**=**"robot\_state\_publisher" type**=**"state\_publisher"**/>**

​

**<!--** Show **in** Rviz **-->**

**<!--<**node name**=**"rviz" pkg**=**"rviz" type**=**"rviz" args**=**"-d $(find my\_mira\_description)/rviz\_config/urdf.rviz"**/>-->**

**<**node name**=**"rviz" pkg**=**"rviz" type**=**"rviz" args**=**""**/>**

​

**</**launch**>**

In this launch file, you have:

In [ ]:



**<**param name**=**"robot\_description" command**=**"cat $(arg model)" **/>**

Here it is loading the URDF file to the param server variable called "robot\_description." Bear in mind that if you are loading more than one robot, you will have to load them in different variables, like "robot1\_description" and "robot2\_description."

In [ ]:



**<!--** send fake joint values **-->**

**<**node name**=**"joint\_state\_publisher" pkg**=**"joint\_state\_publisher" type**=**"joint\_state\_publisher"**>**

**<**param name**=**"use\_gui" value**=**"TRUE"**/>**

**</**node**>**

​

​

**<!--** Combine joint values **-->**

**<**node name**=**"robot\_state\_publisher" pkg**=**"robot\_state\_publisher" type**=**"state\_publisher"**/>**

Start the jointstate publisher and the robotstate publisher. These will publish the TFs of the URDF of the robot links and joints.  
To know more about how this works, please go to our **TF-ROS course**.

In [ ]:



**<!--** Show **in** Rviz **-->**

**<!--<**node name**=**"rviz" pkg**=**"rviz" type**=**"rviz" args**=**"-d $(find my\_mira\_description)/rviz\_config/urdf.rviz"**/>-->**

**<**node name**=**"rviz" pkg**=**"rviz" type**=**"rviz" args**=**""**/>**

Run RVIZ. The part about loading your own RVIZ is commented. The first time you launch this, just save the RVIZ config file and then you will have all that is needed.

|  |
| --- |
| Execute in WebShell #1 |



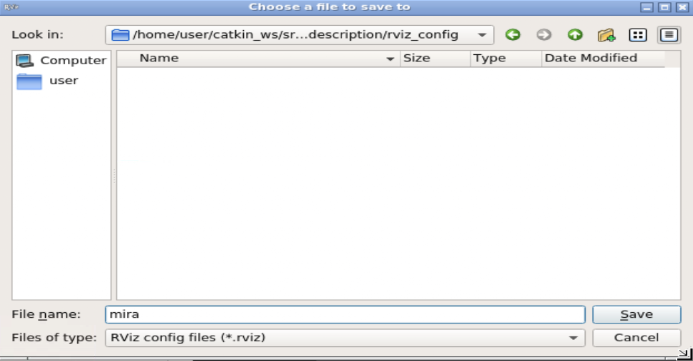
roslaunch my\_mira\_description urdf\_visualize.launch model:='$(find my\_mira\_description)/urdf/mira\_simple.urdf'

This command launches a totally empty RVIZ session, to which you will have to add the TF and RobotModel representations.  
To visualize this, you will have to access the GraphicalTools by pressing the icon:

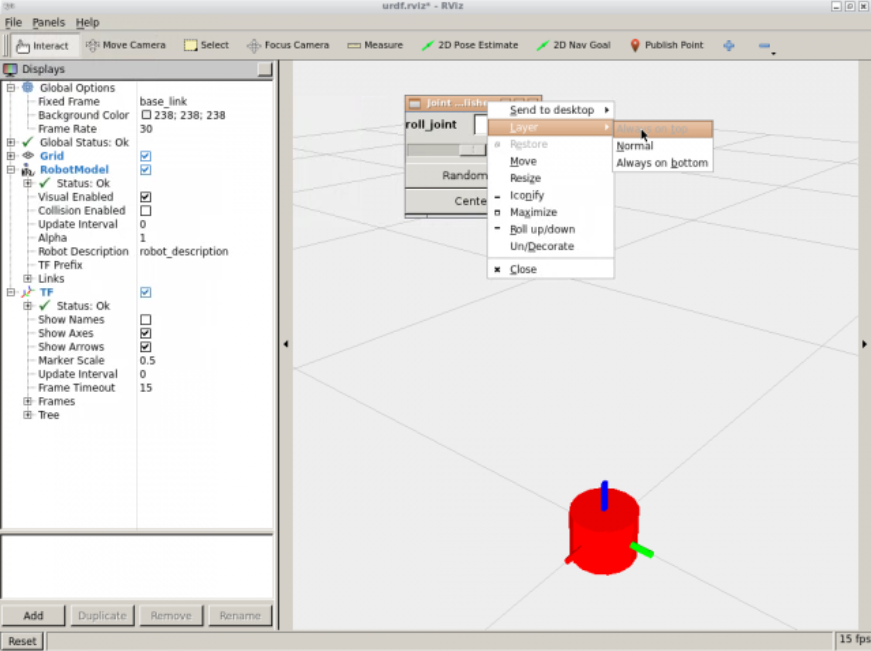


You will have to add two elements in RVIZ:

* RobotModel: In this case, just select the **robot\_description** variable for the **RobotDescription field**.
* TF: It will turn green as soon as you select the **correct fixed frame**, in this case **base\_link**.
* Save the RVIZ file so that you don't have to do this adding every time you launch. Save it in the **rviz\_config folder** you created before with the name **mira**



If all went well, you should have something like this:



Warning

As you can see, you have RVIZ, but also a window with a slider. This slider allows you to move the joints. It's the **JointStatePublisher Gui**.  
This is vital for checking if the joints are correctly set in the URDF. It also allows you to see if the given limits in the joints are the correct ones.  
If **you can't see the "joint control" window**, it must be behind the RVIZ window. Just move it around and, to avoid any further loss, right click on it and select in **Layers-->Always On Top**.

You can also see the Link-Joint structure of any URDF file through the **urdf\_to\_graphiz tool**. Just execute the following:

|  |
| --- |
| Execute in WebShell #1 |

In [ ]:



roscd my\_mira\_description**/**urdf

In [ ]:



urdf\_to\_graphiz mira\_simple.urdf

This tool generates the link and joint tree:



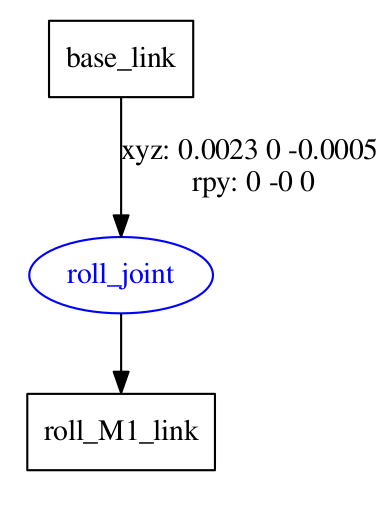
Created file mira\_simple.gv

Created file mira\_simple.pdf

If open the generated **mira.pdf** with the following command, you will get something like this:



evince mira\_simple.pdf



Great, you have your first URDF working!

Exercise U1-1

Play around to get used to the basics of URDF.

* Change the Geometry Shapes
* Change the sizes and investigate how you might be able to see the links inside other links.
* Change the orientation and position of the different links.
* Change the axis of rotation of the joint to make it revolve around Y or Z.
* Change the type of joint to continuous. Bear in mind that you shouldn't have upper and lower limits anymore. But you should leave the effort and speed limits because those are also relevant in continuous joints.
* Change the base\_link colour to orange and the roll\_M1\_link to green.

To change position and orientation, you have to add the origin tag, like so:

In [ ]:



**<?**xml version**=**"1.0"**?>**

**<**robot name**=**"mira"**>**

**<**link name**=**"base\_link"**>**

**<**visual**>**

**<**origin rpy**=**"0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder radius**=**"0.06" length**=**"0.09"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

**</**robot**>**

It's based on the xyz axis of the parent frame; in this case, it's the absolute world frame because it's the first link in the URDF file. **X = RED** color AXIS, **Y = GREEN** color AXIS, and **Z = BLUE** color AXIS.  
As for the rpy (Roll, Pitch, and Yaw ), it's also the parent's axis that corresponds to Roll = Rotation in the X axis, Pitch = Rotation in the Y axis, and Yaw = Rotation in the Z axis.

To change color in URDF, you need to define a material and assign it to the geometry, like so:

In [ ]:



**<?**xml version**=**"1.0"**?>**

**<**robot name**=**"mira"**>**

**<**material name**=**"green"**>**

**<**color rgba**=**"0 0.8 0 1"**/>**

**</**material**>**

​

**<**link name**=**"base\_link"**>**

**<**visual**>**

**<**geometry**>**

**<**cylinder radius**=**"0.06" length**=**"0.09"**/>**

**</**geometry**>**

**<**material name**=**"green"**/>**

**</**visual**>**

**</**link**>**

**</**robot**>**

The color is defined by using the RGBA system, by using a scale of **0 to 1** instead of **0 to 255**. Learn more about RGBA at these example sites.  
<https://www.w3schools.com/css/css3_colors.asp>  
<https://www.w3schools.com/css/tryit.asp?filename=trycss3_color_rgba>

END Exercise U1-1

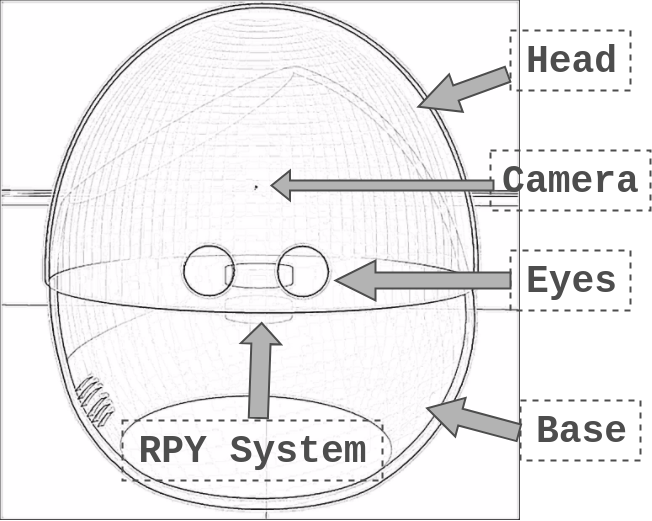
4. Learn About the Morphology of your Robot

The most important step in the creation of a URDF is knowing how the real robot moves.  
You have to decide which type of joints it has and how all of the pieces are linked together.  
In the simulation, you don't always have to mimic the exact way a robot works because you can simplify, as you do not have physical limitations.

Let's have a look at Mira Robot:

The first thing to do is have a look at how it moves. For that, you should have a look at the interview with [Alonso Martinez](https://twitter.com/alonsorobots) on the [Tested YouTube channel](https://youtu.be/0vfuOW1tsX0). Please have a look at how it moves and, more importantly, the mechanism that makes Mira's head have RollPitchYaw capabilities.

Now, have a look at what the parts of Mira's structure are made of:

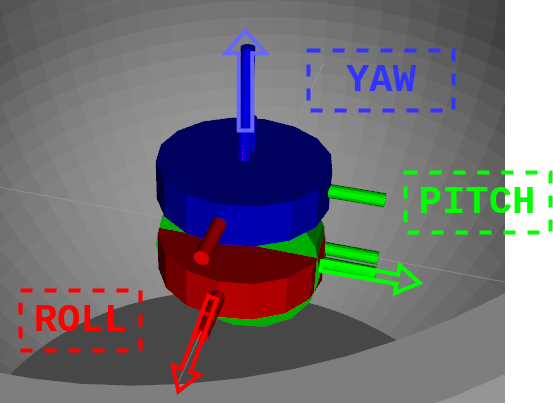


So, it seems that you will need the following links:

* Base\_link
* Head\_link
* Left\_eye\_link and Right\_eye\_link
* Camera\_link
* RPYSystemLink

That's correct, except that we have to define what the RPY system is a bit better. This will also clearly define how the links are connected.  
Let's have a closer look at how the RPY System should look and how it works:

|  |  |  |
| --- | --- | --- |
| Mira | MiraNoHead | MiraNoHead RPYSystem |



This system emulates the real system, giving Roll, Pitch, and Yaw Movements.  
The Roll link connects to the base\_link and rotates around the X axis.  
The Pitch link connects to the base\_link and rotates around the Y axis.  
The Yaw link connects to the base\_link and rotates around the Z axis.  
Using the colors that match the axis helps a lot to keep the rotating axis clear. So bear that in mind when you define these kind of links.  
These joints will also be the ones that are actuated when we introduce the actuators and controls in the simulation.  
Note that we could have positioned the Yaw link in the center, but it's positioned in that way to be easy to see, and to slightly emulate the real system.

Exercise U1-2

Seeing that you now have the structure of Mira clear, generate the whole URDF file, using only geometric shapes to represent the links.  
We recommend that you go step by step. Add one link and a joint, and test. Then, when it's okay, continue to the next one. Otherwise, it can become overwhelming.

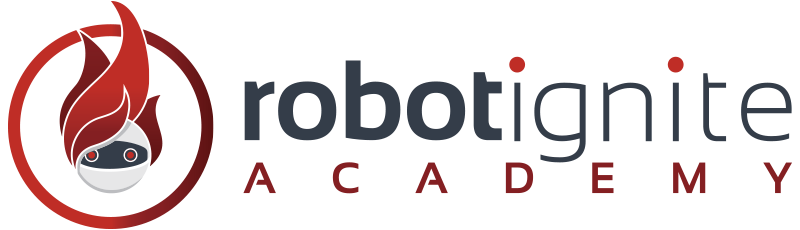
Data for Exercise U1-2

* Joints To be defined:  
  roll\_joint( base\_link with roll\_M1\_link)  
  pitch\_joint( roll\_M1\_link with pitch\_M2\_link)  
  yaw\_joint( pitch\_M2\_link with yaw\_M3\_link)  
  base\_head\_joint( yaw\_M3\_link with head\_link)  
  head\_lefteye\_joint( head\_link with left\_eye\_link)  
  head\_righteye\_joint( head\_link with right\_eye\_link)  
  head\_camera\_joint( head\_link with camera\_link)
* To decide the Joint type, think about whether they have to move or not (fixed), and if they have to be limited (revolute) or not (continuous).
* Here you have the dimensions of the links:  
  base\_link --> cylinder radius="0.06" length="0.09"  
  roll\_M1\_link --> cylinder length="0.005" radius="0.01"  
  pitch\_M2\_link --> cylinder length="0.005" radius="0.01"  
  yaw\_M3\_link --> cylinder length="0.005" radius="0.01"  
  head\_link --> sphere radius="0.06"  
  left\_eye\_link --> cylinder radius="0.00525" length="0.00525"  
  right\_eye\_link --> cylinder radius="0.00525" length="0.00525"  
  camera\_link --> box size="0.0005 0.0005 0.0005"
* You will have to decide the origin of xyz and the RPY of each link. It doesn't have to be perfect, but try your best to experience how difficult it can be to get a model right without a blueprint. You will also get your bearings in the XYZ movement.

END Exercise U1-2

Solution Exercise U1-2

Please try to do it by yourself, unless you get stuck or need some inspiration. You will learn much more if you fight for each exercise.



Here you have the whole URDF file for the Mira model:

In [ ]:



roscd my\_mira\_description

touch urdf**/**mira\_geometric.urdf

mira\_geometric.urdf

In [ ]:



**<?**xml version**=**"1.0"**?>**

**<**robot name**=**"mira"**>**

​

**<**material name**=**"blue"**>**

**<**color rgba**=**"0 0 0.8 1"**/>**

**</**material**>**

**<**material name**=**"red"**>**

**<**color rgba**=**"0.8 0 0 1"**/>**

**</**material**>**

**<**material name**=**"green"**>**

**<**color rgba**=**"0 0.8 0 1"**/>**

**</**material**>**

**<**material name**=**"grey"**>**

**<**color rgba**=**"0.75 0.75 0.75 1"**/>**

**</**material**>**

**<**material name**=**"white"**>**

**<**color rgba**=**"1.0 1.0 1.0 1"**/>**

**</**material**>**

**<**material name**=**"black"**>**

**<**color rgba**=**"0 0 0 1"**/>**

**</**material**>**

​

**<!--** **\*** **\*** **\*** Link Definitions **\*** **\*** **\*** **-->**

**<**link name**=**"base\_link"**>**

​

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder radius**=**"0.06" length**=**"0.09"**/>**

**</**geometry**>**

**<**material name**=**"grey"**/>**

**</**visual**>**

**</**link**>**

​

​

​

**<**link name**=**"roll\_M1\_link"**>**

​

**<**visual**>**

**<**origin rpy**=**"0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder length**=**"0.005" radius**=**"0.01"**/>**

**</**geometry**>**

**<**material name**=**"red"**/>**

**</**visual**>**

**</**link**>**

​

**<**joint name**=**"roll\_joint" type**=**"revolute"**>**

**<**parent link**=**"base\_link"**/>**

**<**child link**=**"roll\_M1\_link"**/>**

**<**origin xyz**=**"0.0023 0 -0.0005" rpy**=**"0 0 0"**/>**

**<**limit lower**=**"-0.2" upper**=**"0.2" effort**=**"0.1" velocity**=**"0.005"**/>**

**<**axis xyz**=**"1 0 0"**/>**

**</**joint**>**

​

​

​

**<**link name**=**"pitch\_M2\_link"**>**

​

**<**visual**>**

**<**origin rpy**=**"0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder length**=**"0.005" radius**=**"0.01"**/>**

**</**geometry**>**

**<**material name**=**"green"**/>**

**</**visual**>**

**</**link**>**

​

​

**<**joint name**=**"pitch\_joint" type**=**"revolute"**>**

**<**parent link**=**"roll\_M1\_link"**/>**

**<**child link**=**"pitch\_M2\_link"**/>**

**<**origin xyz**=**"0 0 0" rpy**=**"0 -1.5708 0"**/>**

**<**limit lower**=**"0" upper**=**"0.44" effort**=**"0.1" velocity**=**"0.005"**/>**

**<**axis xyz**=**"0 1 0"**/>**

**</**joint**>**

​

​

**<**link name**=**"yaw\_M3\_link"**>**

​

**<**visual**>**

**<**origin rpy**=**"0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder length**=**"0.005" radius**=**"0.01"**/>**

**</**geometry**>**

**<**material name**=**"blue"**/>**

**</**visual**>**

**</**link**>**

​

**<**joint name**=**"yaw\_joint" type**=**"continuous"**>**

**<**parent link**=**"pitch\_M2\_link"**/>**

**<**child link**=**"yaw\_M3\_link"**/>**

**<**origin xyz**=**"0.01 0 0" rpy**=**"0 1.5708 0"**/>**

**<**limit effort**=**"0.1" velocity**=**"0.01"**/>**

**<**axis xyz**=**"0 0 1"**/>**

**</**joint**>**

​

​

**<**link name**=**"head\_link"**>**

​

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**sphere radius**=**"0.06"**/>**

**</**geometry**>**

**<**material name**=**"white"**/>**

**</**visual**>**

**</**link**>**

​

​

**<**joint name**=**"base\_head\_joint" type**=**"fixed"**>**

**<**parent link**=**"yaw\_M3\_link"**/>**

**<**child link**=**"head\_link"**/>**

**<**origin xyz**=**"0 0 0.06" rpy**=**"0 0 0"**/>**

**</**joint**>**

​

**<**link name**=**"left\_eye\_link"**>**

​

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder radius**=**"0.00525" length**=**"0.00525"**/>**

**</**geometry**>**

**<**material name**=**"black"**/>**

**</**visual**>**

**</**link**>**

​

**<**link name**=**"right\_eye\_link"**>**

​

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder radius**=**"0.00525" length**=**"0.00525"**/>**

**</**geometry**>**

**<**material name**=**"black"**/>**

**</**visual**>**

**</**link**>**

​

**<**joint name**=**"head\_lefteye\_joint" type**=**"fixed"**>**

**<**parent link**=**"head\_link"**/>**

**<**child link**=**"left\_eye\_link"**/>**

**<**origin xyz**=**"0.0095 0.057 0.0085" rpy**=**"-1.5708 0 0"**/>**

**</**joint**>**

​

**<**joint name**=**"head\_righteye\_joint" type**=**"fixed"**>**

**<**parent link**=**"head\_link"**/>**

**<**child link**=**"right\_eye\_link"**/>**

**<**origin xyz**=**"-0.0095 0.057 0.0085" rpy**=**"-1.5708 0 0"**/>**

**</**joint**>**

​

**<**link name**=**"camera\_link"**>**

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**box size**=**"0.0005 0.0005 0.0005"**/>**

**</**geometry**>**

**<**material name**=**"green"**/>**

**</**visual**>**

**</**link**>**

​

​

**<**joint name**=**"head\_camera\_joint" type**=**"fixed"**>**

**<**parent link**=**"head\_link"**/>**

**<**child link**=**"camera\_link"**/>**

**<**origin xyz**=**"0 0.057 0.0255" rpy**=**"0 0 0"**/>**

**</**joint**>**

​

​

**</**robot**>**

END mira\_geometric.urdf

If we now execute the visualization launch for urdfs , you should have something similar to this:



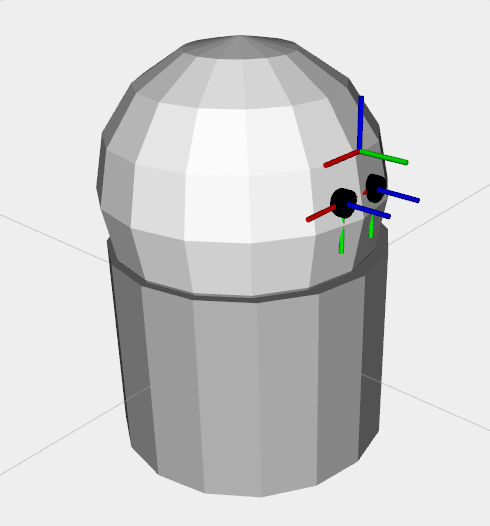
roslaunch my\_mira\_description urdf\_visualize.launch model:='$(find my\_mira\_description)/urdf/mira\_geometric.urdf'

Errors and warnings that could appear, which have no effect and normal in this ROS distribution:



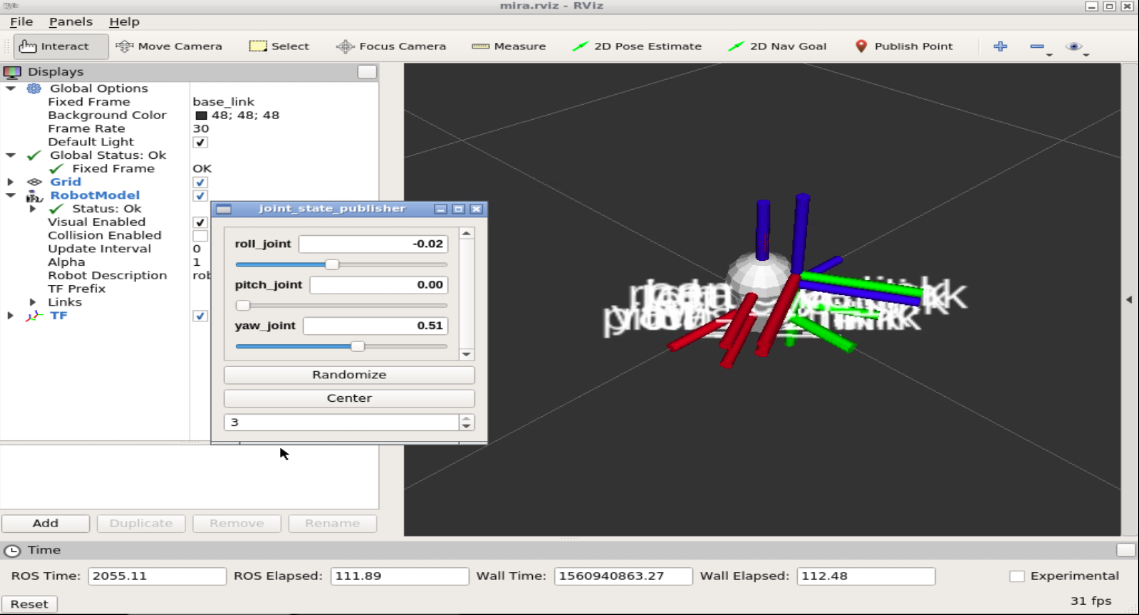
[WARN] [1592297719.572633, 1233.241000]: The 'use\_gui' parameter was specified, which is deprecated. We'll attempt to find and run the GUI, but if this fails you should install the 'joint\_state\_publisher\_gui' package instead and run that. This backwards compatibility option will be removed in Noetic.

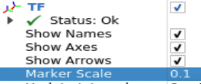
Remember to open the **mira.rviz** file you saved previously in the **rviz\_config folder**.



To get this visualization , you will have to:

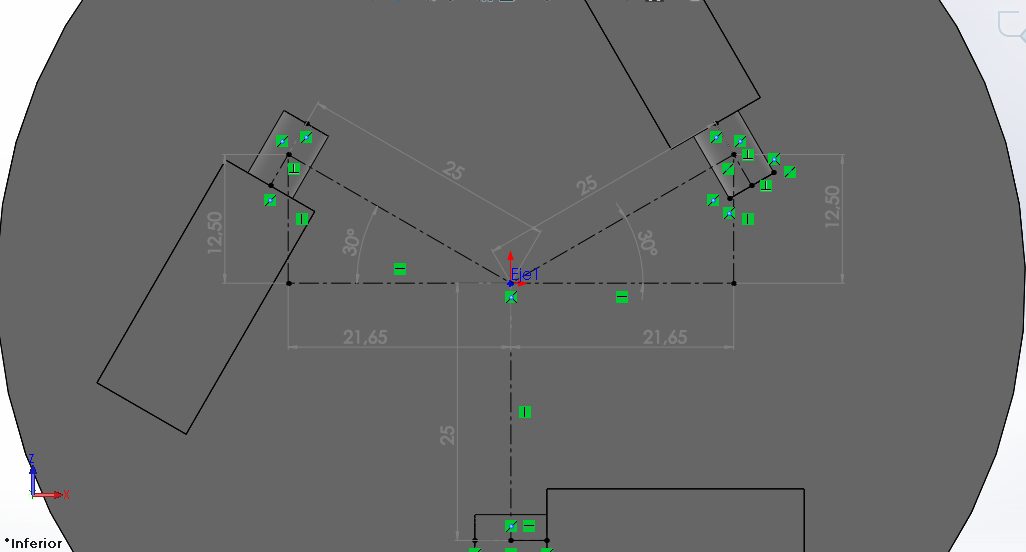
* Decrease the size of the Marker scale from 1.0 to 0.1.
* Uncheck the TF tree if you only want to see the model.







Did you manage to do something similar to this?  
Naturally, one of the most difficult aspects was probably positioning the elements.  
That's why it's so important to iterate step by step, and if possible, have all of the assembly measurements in blueprints, like this one:



This was taken from the Solid Works model of the Gurdy Robot that you will create later on in the course.  
It's highly advisable to have a CAD model fully assembled to be able to get precise measurements on how the joints have to be positioned. Otherwise, it's an iterative slow process, as you have experienced.

5. Learn how to import your 3D CAD models to Gazebo

This is a very wide topic and there are many ways of doing it. Here you will learn just the basic elements needed for correctly importing 3D models.

* When using a CAD tool, the final models have to be saved in STL format or Dae format. SolidWorks, for example, gives the option of STL. The only advantage with DAE is that DAE saves color information. But that's something that can be added afterwards in Blender.
* Import the STL of the DAE models into Blender. Here you will have to set the origin of the model and the units. This is absolutely vital. The setting of the axis will determine how difficult it is to create the URDF model. As for the units, having a 10-foot high Mira Robot is not very realistic, apart from the effects on the inertias afterwards.
* Once you have the axis and units set up, you can add a material to the blender model to add color.
* Once done, you export it to dae format. Sometimes, blender doesn't save properly the first time, so you will have to import the new dae into blender again in an empty scene and then export it again to dae.

To know how to import a model from a CAD system to blender and set the origin, please have a look at our video tutorial on this matter:<https://youtu.be/aP4sDyrRzpU>  
To correctly set the units in blender and other CAD techniques in Blender, this site is great: <http://www.rab3d.com/tut_blen_2-6x_608-1.php>  
To add materials to blender, please refer to this tutorial: <https://youtu.be/rRdKj33Keec>

In this case, you will have already been provided with the models needed for this URDF:

|  |
| --- |
| Execute in WebShell #1 |

In [ ]:



roscd mira\_description

In [ ]:



cp .**/**models**/**mira**/**meshes**/**mira\_body\_v3.dae **/**home**/**user**/**catkin\_ws**/**src**/**

In [ ]:



cp .**/**models**/**mira**/**meshes**/**mira\_head\_v5.dae **/**home**/**user**/**catkin\_ws**/**src**/**

In [ ]:



cp .**/**models**/**mira**/**meshes**/**mira\_eye\_v4.dae **/**home**/**user**/**catkin\_ws**/**src**/**

In [ ]:



roscd my\_mira\_description

In [ ]:



cd models;mkdir mira;cd mira;mkdir meshes

In [ ]:



cd **/**home**/**user**/**catkin\_ws**/**src**/**;mv **\***.dae **/**home**/**user**/**catkin\_ws**/**src**/**my\_mira\_description**/**models**/**mira**/**meshes**/**

And now, place them inside the **my\_mira\_description/models/mira/meshes/**

Finally, you just have to replace the geometry that are now spheres, cylinders, and so on by the .dae files. Here is an example of how it would be done in the base\_link case:

In [ ]:



**<**link name**=**"base\_link"**>**

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**mesh filename**=**"package://my\_mira\_description/models/mira/meshes/mira\_body\_v3.dae"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

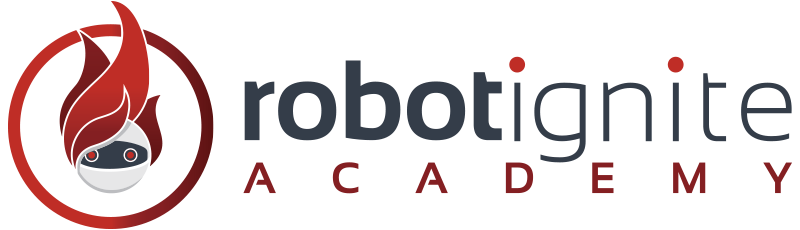
Exercise U1-3

Replace the head, body, and eyes by their 3Dmeshes.  
You will probably need to make some changes in the joint positions and orientations.

END Exercise U1-3

Solution Exercise U1-3

Please try to do it by yourself, unless you get stuck or need some inspiration. You will learn much more if you fight for each exercise.



You should get the full model. Check that all of the limits are correct and that nothing is out of place. A good practice is to put all of the joints in maximum and minimum positions, to see if everything is okay.



roscd my\_mira\_description

touch urdf/mira\_only\_daes.urdf

mira\_only\_daes.urdf

In [ ]:



**<?**xml version**=**"1.0"**?>**

**<**robot name**=**"mira"**>**

​

**<**material name**=**"blue"**>**

**<**color rgba**=**"0 0 0.8 1"**/>**

**</**material**>**

**<**material name**=**"red"**>**

**<**color rgba**=**"0.8 0 0 1"**/>**

**</**material**>**

**<**material name**=**"green"**>**

**<**color rgba**=**"0 0.8 0 1"**/>**

**</**material**>**

​

**<!--** **\*** **\*** **\*** Link Definitions **\*** **\*** **\*** **-->**

**<**link name**=**"base\_link"**>**

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**mesh filename**=**"package://my\_mira\_description/models/mira/meshes/mira\_body\_v3.dae"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

​

​

​

**<**link name**=**"roll\_M1\_link"**>**

​

**<**visual**>**

**<**origin rpy**=**"0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder length**=**"0.005" radius**=**"0.01"**/>**

**</**geometry**>**

**<**material name**=**"red"**/>**

**</**visual**>**

**</**link**>**

**<!--** This **is** **for** color **and** physical properties **in** Gazebo, color won't work with the material tag in gazebo

only **for** URDF coloring **-->**

**<**gazebo reference**=**"roll\_M1\_link"**>**

**<**kp**>**1000.0**</**kp**>**

**<**kd**>**10.0**</**kd**>**

**<**mu1**>**10.0**</**mu1**>**

**<**mu2**>**10.0**</**mu2**>**

**<**material**>**Gazebo**/**Red**</**material**>**

**</**gazebo**>**

​

**<**joint name**=**"roll\_joint" type**=**"revolute"**>**

**<**parent link**=**"base\_link"**/>**

**<**child link**=**"roll\_M1\_link"**/>**

**<**origin xyz**=**"0.0023 0 -0.0005" rpy**=**"0 0 0"**/>**

**<**limit lower**=**"-0.2" upper**=**"0.2" effort**=**"0.1" velocity**=**"0.005"**/>**

**<**axis xyz**=**"1 0 0"**/>**

**</**joint**>**

​

​

**<**link name**=**"pitch\_M2\_link"**>**

​

**<**visual**>**

**<**origin rpy**=**"0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder length**=**"0.005" radius**=**"0.01"**/>**

**</**geometry**>**

**<**material name**=**"green"**/>**

**</**visual**>**

**</**link**>**

​

**<!--** This **is** **for** color **and** physical properties **in** Gazebo, color won't work with the material tag in gazebo

only **for** URDF coloring **-->**

**<**gazebo reference**=**"pitch\_M2\_link"**>**

**<**kp**>**1000.0**</**kp**>**

**<**kd**>**10.0**</**kd**>**

**<**mu1**>**10.0**</**mu1**>**

**<**mu2**>**10.0**</**mu2**>**

**<**material**>**Gazebo**/**Green**</**material**>**

**</**gazebo**>**

​

**<**joint name**=**"pitch\_joint" type**=**"revolute"**>**

**<**parent link**=**"roll\_M1\_link"**/>**

**<**child link**=**"pitch\_M2\_link"**/>**

**<**origin xyz**=**"0 0 0" rpy**=**"0 -1.5708 0"**/>**

**<**limit lower**=**"0" upper**=**"0.44" effort**=**"0.1" velocity**=**"0.005"**/>**

**<**axis xyz**=**"0 1 0"**/>**

**</**joint**>**

​

​

​

**<**link name**=**"yaw\_M3\_link"**>**

​

**<**visual**>**

**<**origin rpy**=**"0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder length**=**"0.005" radius**=**"0.01"**/>**

**</**geometry**>**

**<**material name**=**"blue"**/>**

**</**visual**>**

**</**link**>**

​

**<!--** This **is** **for** color **and** physical properties **in** Gazebo, color won't work with the material tag in gazebo

only **for** URDF coloring **-->**

**<**gazebo reference**=**"yaw\_M3\_link"**>**

**<**kp**>**1000.0**</**kp**>**

**<**kd**>**10.0**</**kd**>**

**<**mu1**>**10.0**</**mu1**>**

**<**mu2**>**10.0**</**mu2**>**

**<**material**>**Gazebo**/**Blue**</**material**>**

**</**gazebo**>**

​

​

**<**joint name**=**"yaw\_joint" type**=**"continuous"**>**

**<**parent link**=**"pitch\_M2\_link"**/>**

**<**child link**=**"yaw\_M3\_link"**/>**

**<**origin xyz**=**"0.01 0 0" rpy**=**"0 1.5708 0"**/>**

**<**limit effort**=**"0.1" velocity**=**"0.01"**/>**

**<**axis xyz**=**"0 0 1"**/>**

**</**joint**>**

​

​

**<**link name**=**"head\_link"**>**

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**mesh filename**=**"package://my\_mira\_description/models/mira/meshes/mira\_head\_v5.dae"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

​

​

**<**joint name**=**"base\_head\_joint" type**=**"fixed"**>**

**<**parent link**=**"yaw\_M3\_link"**/>**

**<**child link**=**"head\_link"**/>**

**<**origin xyz**=**"0 0 0" rpy**=**"0 0 0"**/>**

**</**joint**>**

​

**<**link name**=**"left\_eye\_link"**>**

​

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<!--**

**<**cylinder radius**=**"0.00525" length**=**"0.00525"**/>**

**-->**

**<**mesh filename**=**"package://my\_mira\_description/models/mira/meshes/mira\_eye\_v4.dae"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

​

**<**link name**=**"right\_eye\_link"**>**

​

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<!--**

**<**cylinder radius**=**"0.00525" length**=**"0.00525"**/>**

**-->**

**<**mesh filename**=**"package://my\_mira\_description/models/mira/meshes/mira\_eye\_v4.dae"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

​

**<**joint name**=**"head\_lefteye\_joint" type**=**"fixed"**>**

**<**parent link**=**"head\_link"**/>**

**<**child link**=**"left\_eye\_link"**/>**

**<**origin xyz**=**"0.0095 0.057 0.0085" rpy**=**"-1.5708 0 0"**/>**

**</**joint**>**

​

**<**joint name**=**"head\_righteye\_joint" type**=**"fixed"**>**

**<**parent link**=**"head\_link"**/>**

**<**child link**=**"right\_eye\_link"**/>**

**<**origin xyz**=**"-0.0095 0.057 0.0085" rpy**=**"-1.5708 0 0"**/>**

**</**joint**>**

​

**<**link name**=**"camera\_link"**>**

​

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**box size**=**"0.0005 0.0005 0.0005"**/>**

**</**geometry**>**

**<**material name**=**"green"**/>**

**</**visual**>**

**</**link**>**

​

**<**gazebo reference**=**"camera\_link"**>**

**<**material**>**Gazebo**/**Green**</**material**>**

**</**gazebo**>**

​

**<**joint name**=**"head\_camera\_joint" type**=**"fixed"**>**

**<**parent link**=**"head\_link"**/>**

**<**child link**=**"camera\_link"**/>**

**<**origin xyz**=**"0 0.057 0.0255" rpy**=**"0 0 0"**/>**

**</**joint**>**

​

​

​

​

**</**robot**>**

END mira\_only\_daes.urdf

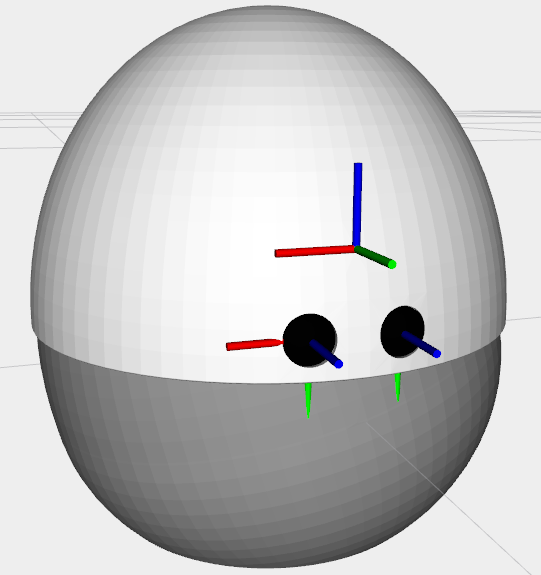
If we now execute the visualization launch for urdfs , you should have something similar to this:

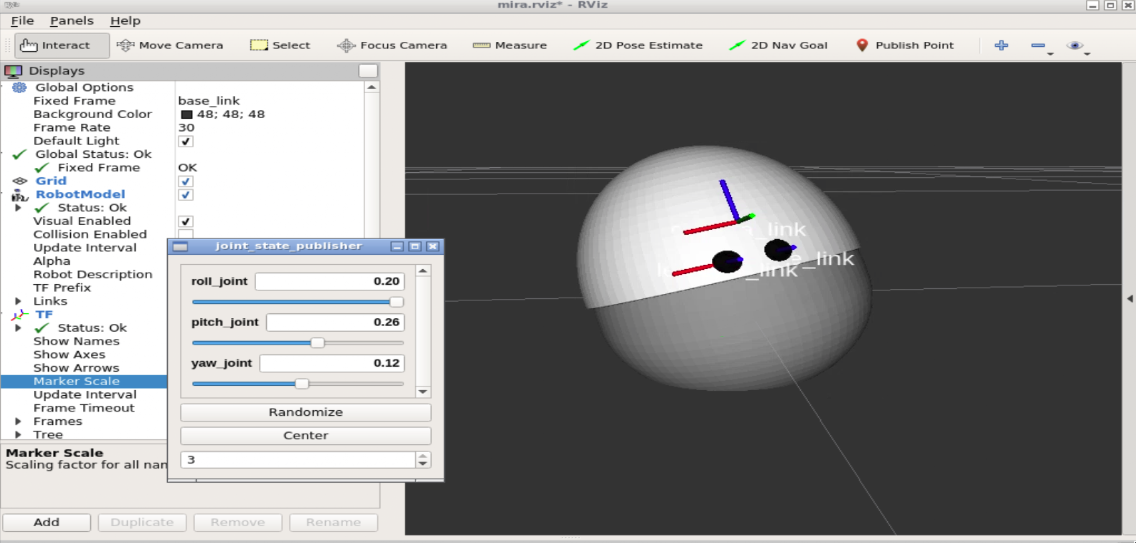


roslaunch my\_mira\_description urdf\_visualize.launch model:='$(find my\_mira\_description)/urdf/mira\_only\_daes.urdf'

Remember to open the **mira.rviz** file you saved previously in the **rviz\_config folder**.

You should have something like this:





For further details on all the options availabe in URDF, please refer to this link: <http://wiki.ros.org/urdf/XML/joint>

Congratulations! You are now ready to learn how to spawn this URDF file into Gazebo Simulator. Please proceed to Unit 2.