## Docker + ROS = URDF

Движок PyBullet поддерживает различные форматы объектов и даже может создавать их на лету в процессе работы, но нас в первую очередь будет интересовать формат URDF (Unified Robotics Description Format). Он широко применяется для промышленной и исследовательской разработки моделей роботов. Несмотря на солидное название, по сути это просто xml-файл, в котором с помощью тегов описываются детали (links) и связи между ними (joints). Например, простой цилиндрический робот в вакууме может выглядеть вот так:

<?xml version="1.0"?>

<robot name="myfirst">

<link name="base\_link">

<visual>

<geometry>

<cylinder length="0.6" radius="0.2"/>

</geometry>

</visual>

</link>

</robot>

Чтобы определить деталь, необходимо задать систему координат, визуальную геометрию и физические свойства. Для соединения нужно указать тип и физические характеристики. Для старта рекомендую начать со статей

<https://articulatedrobotics.xyz/ready-for-ros-6-tf/>

<https://articulatedrobotics.xyz/ready-for-ros-7-urdf/>

## ROS (Robotic operating system)

Чтобы создавать и отлаживать URDF-модели нам понадобится специальная среда - ROS (Robotic Operation System). Она предназначена, как можно предположить из названия, для разработки и развертывания робототехнических систем на различных вычислительных платформах. ROS содержит в себе огромное количество разных подсистем и инструментов, но нам в ближайшее время потребуется только средство визуализации моделей rviz (кстати, если вы найдете какой-то работоспособный редактор URDF-моделей, независимый от ROS, дайте мне знать ^\_^).

Отдельный вопрос состоит в том, как запустить ROS и воспользоваться нужными компонентами. На официальном сайте представлены несколько вариантов, но наиболее простым я считаю запуска ROS в Docker-контейнере.

Примечание: когда вы начнёте читать документацию, то увидите, что в настоящее время есть 2 версии ROS: собственно ROS и ROS 2. Для наших задач подойдёт любая, но чтобы снизить неопределенность я буду иметь в виду стабильный дистрибутив ROS2 Humble Hawksbill.

## Docker

Docker – это отличный платформа для виртуализации на уровне операционной системы. Он позволяет создавать независимые окружения для запуска приложений и сервисов. Поэтому предлагаю воспользоваться его мощью и запустить официальный образ разработчиков ROS. За любыми подробностями о Docker отсылаю к циклу статей на Хабре: <https://habr.com/en/company/ruvds/blog/440660/>

## Подготовка окружения

Подготовка и запуск окружения Docker + ROS 2 описана для Ubuntu Linux (20.04+). При желании можно отказаться от Docker и просто установить ROS 2 так, как это описано в документации.

1. Устанавливаем Docker
2. Загружаем нужный образ

docker pull osrf/ros:humble-desktop-full

1. Запускаем контейнер

docker run -it \

--env="DISPLAY" \

--env="QT\_X11\_NO\_MITSHM=1" \

--volume="/tmp/.X11-unix:/tmp/.X11-unix:rw" \

osrf/ros:humble-desktop-full

1. Чтобы запускать графические приложения (в частности rviz), необходимо пробросить из докера соединение с X-сервером

xhost +local:container\_hostname

где в качестве container\_hostname в нашем случае подойдет ID контейнера (потому что мы не настраивали hostname)

1. Теперь контейнер можно запустить

docker start containerId

1. И продключиться к нему

docker container attach containerId

Доброе пожаловать в среду ROS2 ;)

## Список материалов

Docker:

<https://habr.com/en/company/ruvds/blog/440660/>

ROS + Docker

<http://docs.ros.org/en/humble/How-To-Guides/Run-2-nodes-in-single-or-separate-docker-containers.html>

ROS + Docker + X11

<http://wiki.ros.org/docker/Tutorials/GUI>

URDF:

<http://docs.ros.org/en/humble/Tutorials/URDF/Building-a-Visual-Robot-Model-with-URDF-from-Scratch.html>

Дополнительные материалы

<http://wiki.ros.org/urdf/Tutorials>