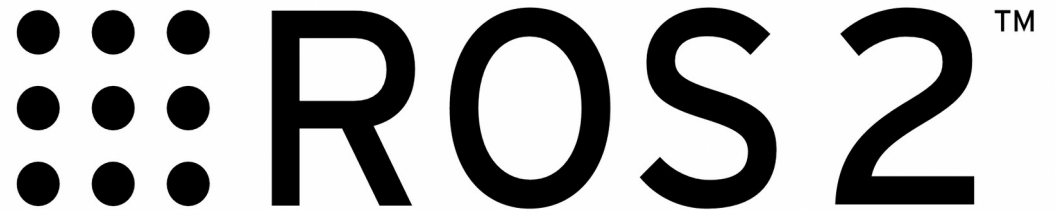


# Практикум 1

Введение в ROS

# Robot operation system

- ROS – операционная система роботов.
- Набор библиотек, включающий инструменты для решения задач робототехники.
- Призвана решить задачу повторяющегося “изобретения велосипеда”.



# Useful programmms



Vscode- для кодинга



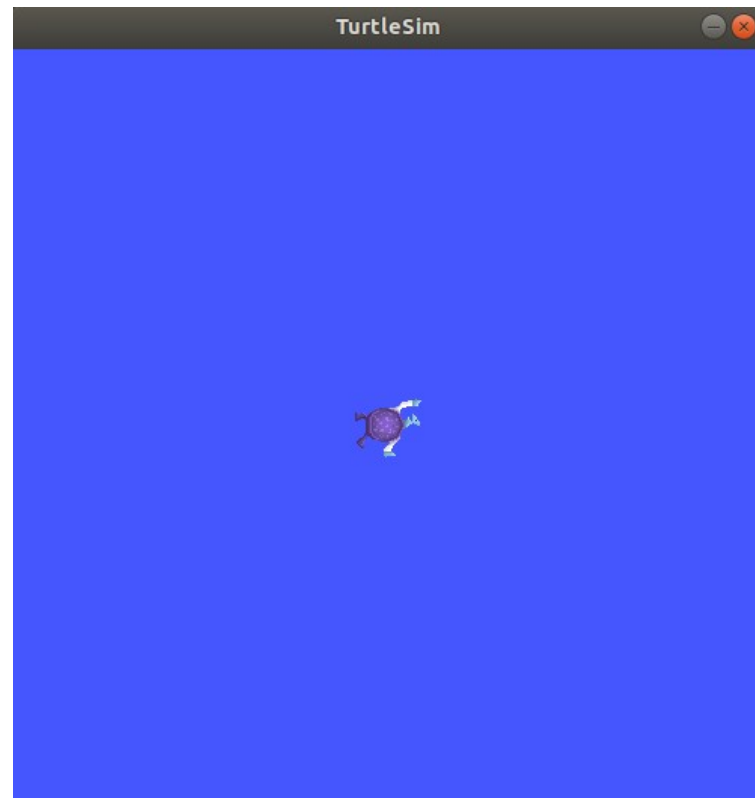
Docker – для запуска приложений  
Linux Ha windows



MS Edge –  
для бесплатного  
chatgpt5

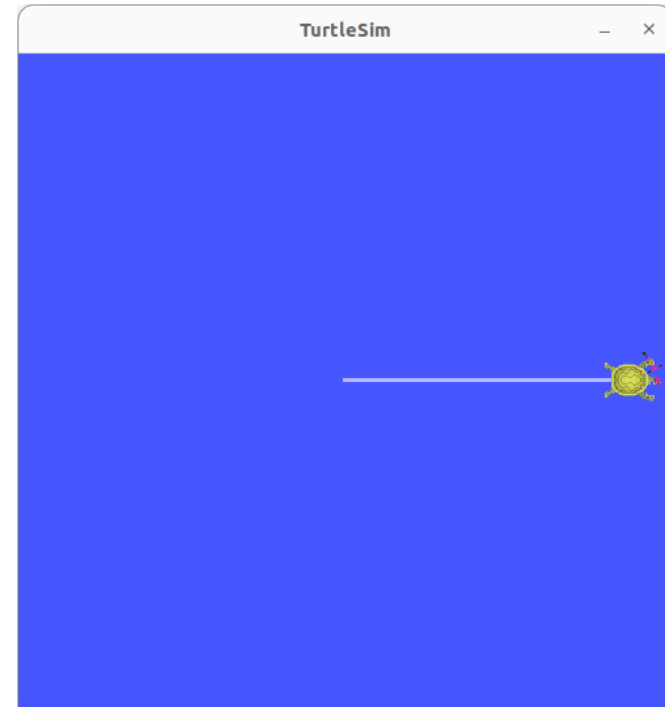
# Turtlesim (running the node)

```
temirlan@temirlan-Legion-Y7000P-IRX9: ~  
temirlan@temirlan-Legion-Y7000P-IRX9:~$ ros2 run turtlesim turtlesim_node  
[INFO] [1758165964.251713653] [turtlesim]: Starting turtlesim with node name /turtlesim  
[INFO] [1758165964.258509739] [turtlesim]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]  
█
```



# Turtlesim(moving the robot)

```
temirlan@temirlan-Legion-Y7000P-IRX9: ~  
temirlan@temirlan-Le... x temirlan@temirlan-Le... x temirlan@temirlan-Le... x  
temirlan@temirlan-Legion-Y7000P-IRX9:~$ ros2 run turtlesim turtle_teleop_key  
Reading from keyboard  
-----  
Use arrow keys to move the turtle.  
Use G|B|V|C|D|E|R|T keys to rotate to absolute orientations. 'F' to cancel a rotation.  
'Q' to quit.  
█
```



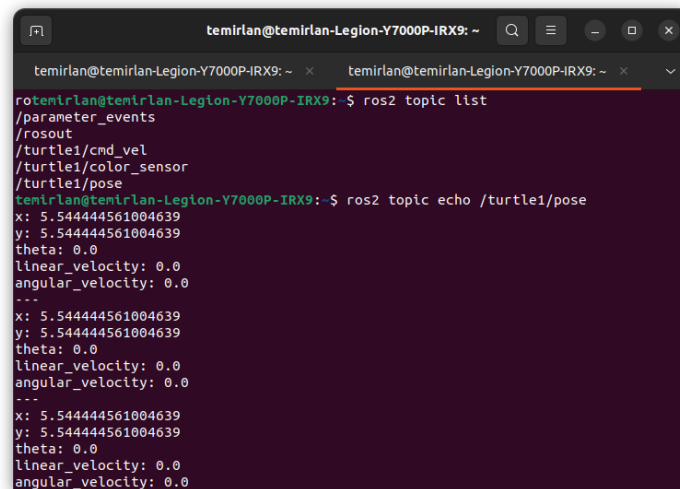
# Turtlesim (check topics)

## Note

Pressing an arrow key will only cause the turtle to move a short distance and then stop. This is because, realistically, you wouldn't want a robot to continue carrying on an instruction if, for example, the operator lost the connection to the robot.

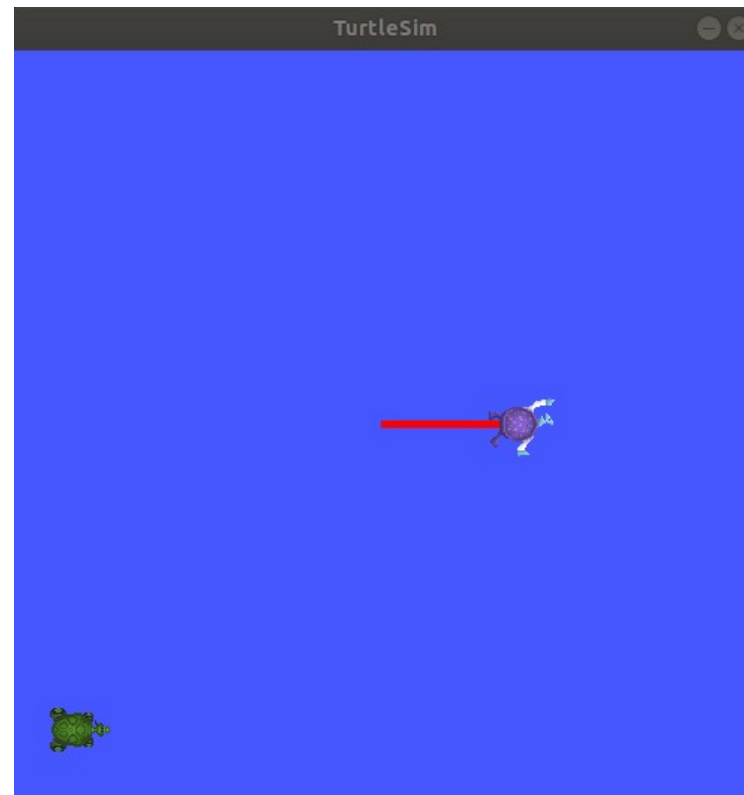
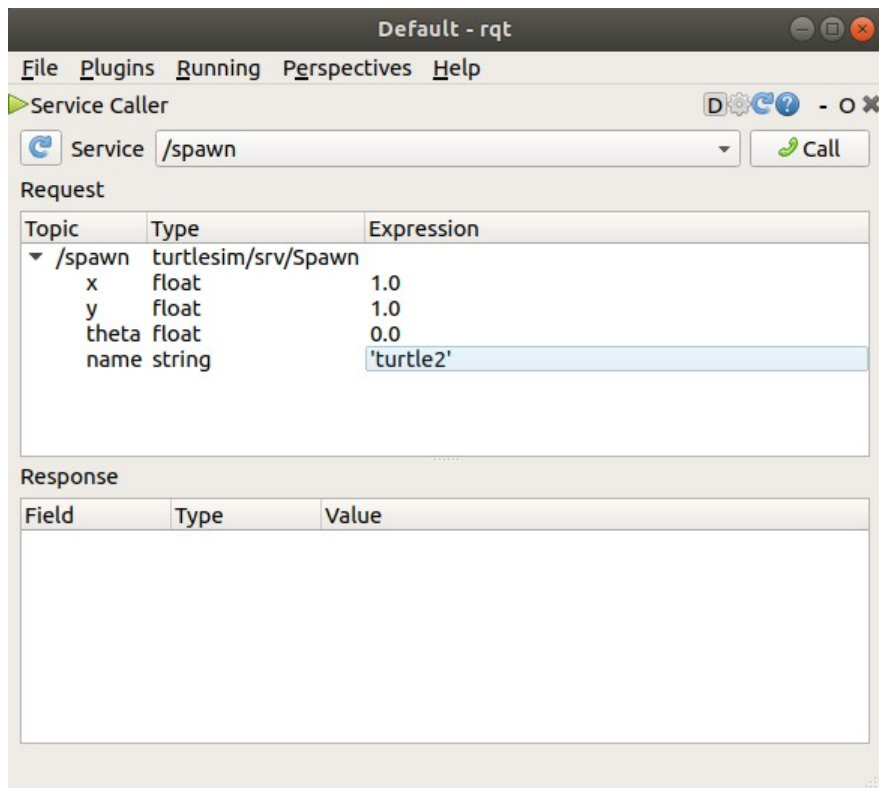
You can see the nodes, and their associated topics, services, and actions, using the `list` subcommands of the respective commands:

```
$ ros2 node list
$ ros2 topic list
$ ros2 service list
$ ros2 action list
```



```
temirlan@temirlan-Legion-Y7000P-IRX9: ~  
temirlan@temirlan-Legion-Y7000P-IRX9: ~  
temirlan@temirlan-Legion-Y7000P-IRX9: ~$ ros2 topic list  
/parameter_events  
/rosout  
/turtle1/cmd_vel  
/turtle1/color_sensor  
/turtle1/pose  
temirlan@temirlan-Legion-Y7000P-IRX9: ~$ ros2 topic echo /turtle1/pose  
x: 5.544444561004639  
y: 5.544444561004639  
theta: 0.0  
linear_velocity: 0.0  
angular_velocity: 0.0  
---  
x: 5.544444561004639  
y: 5.544444561004639  
theta: 0.0  
linear_velocity: 0.0  
angular_velocity: 0.0  
---  
x: 5.544444561004639  
y: 5.544444561004639  
theta: 0.0  
linear_velocity: 0.0  
angular_velocity: 0.0
```

# Turtlesim(spawning a new turtle)



# Turtlesim(setting the colour of the pen)

Default - rqt

File Plugins Running Perspectives Help

Service Caller

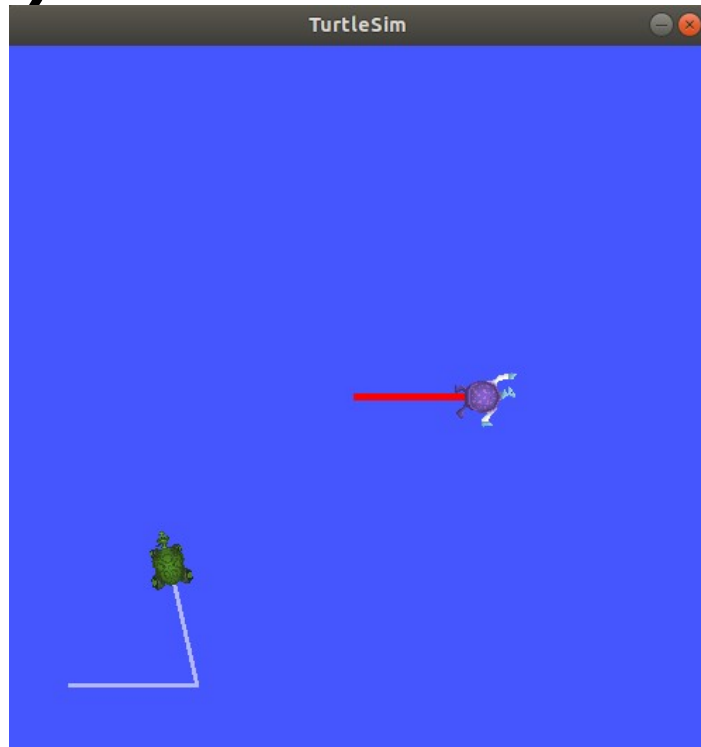
Service: /turtle1/set\_pen Call

Request

Topic	Type	Expression
▼ /turtle1/set_pen	turtlesim/srv/SetPen	
r	uint8	255
g	uint8	0
b	uint8	0
width	uint8	5
off	uint8	0

Response

Field	Type	Value
-------	------	-------

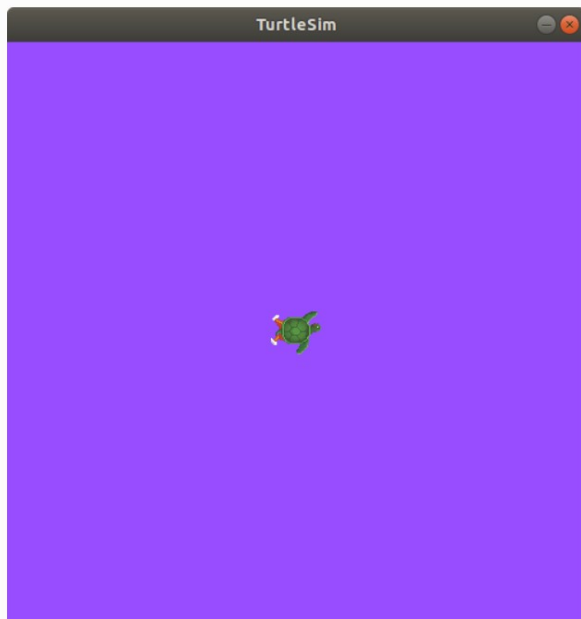




# Turtlesim(changing the background colour)

```
$ ros2 param set /turtlesim background_r 150  
Set parameter successful
```

The background of your turtlesim window should change colors:



# Homework1

- Создать гитхаб аккаунт.
- Заполнить фео и аккаунт в таблицу Google spreadsheets, чтобы добавиться в репозиторий (старостам создать google spreadsheets)
- Зайти в репозиторий
- Установить Docker Desktop, следуя инструкции в репозитории.
- По инструкции в гитхабе, скачать chapter1, и запустить turtlesim
- Отметиться, удалось ли запустить.



# Homework2(spoiler)

- Написать publisher и subscriber, которые будут выводить:
  - 1. Позу робота в реальном времени с помощью степеней свободы
  - 1. Позу робота с помощью матрицы
  - 2. Позу робота относительно центральной системы координат
  - 3. Позу робота относительно верхнего левого угла



# Тutorials

- <https://docs.ros.org/en/humble/Tutorials.html>
- <https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Introducing-Turtlesim/Introducing-Turtlesim.html>

- Beginner: CLI tools

- Configuring environment
- Using `turtlesim`, `ros2`, and `rqt`
- Understanding nodes
- Understanding topics
- Understanding services
- Understanding parameters
- Understanding actions
- Using `rqt_console` to view logs
- Launching nodes
- Recording and playing back data

- Beginner: Client libraries

- Using `colcon` to build packages
- Creating a workspace
- Creating a package
- Writing a simple publisher and subscriber (C++)
- Writing a simple publisher and subscriber (Python)
- Writing a simple service and client (C++)
- Writing a simple service and client (Python)