# Lab2 deliverable

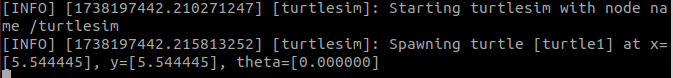
## Task2

Q1:

A close-up of a computer screen

Description automatically generated

Q2:



Q3:

It would blocked by the edge of the window

With warning “Oh I hit the wall”

A screenshot of a video game

Description automatically generated

Q4:

2 node

Q5:

A screenshot of a computer screen

Description automatically generated

The `/turtlesim` node is responsible for simulating the movement of the turtle.

- \*\*Subscribed topics:\*\*

- `/turtle1/cmd\_vel` (message type: `geometry\_msgs/msg/Twist`): Receives velocity commands to move the turtle.

- \*\*Published topics:\*\*

- `/turtle1/pose` (message type: `turtlesim/msg/Pose`): Publishes the turtle's current position (`x, y, theta`).

- `/turtle1/color\_sensor` (message type: `turtlesim/msg/Color`): Publishes the color of the pixel under the turtle.

These topics allow the turtle to receive movement commands and provide feedback about its position and environment.

Q6：

The `/teleop\_turtle` node is responsible for controlling the turtle's movement using keyboard inputs.

- \*\*Subscribed topics:\*\* None (it does not directly subscribe to any topics; instead, it takes user input from the keyboard).

- \*\*Published topics:\*\*

- `/turtle1/cmd\_vel` (message type: `geometry\_msgs/msg/Twist`): Sends velocity commands to move the turtle.

This means `/teleop\_turtle` translates user keyboard input into velocity commands, which are then received by `/turtlesim` to move the turtle.

Q7:

A diagram of a program

Description automatically generated

Yes the arrows match the information before.

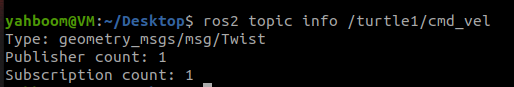
Q8:

A screen shot of a computer

Description automatically generated

List -t provide the type of the topic

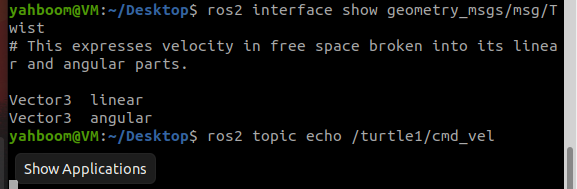
Q9:



The command provides information about the **/turtle1/cmd\_vel** topic, including:

* **Message type**: The type of message being used for this topic.
* **Publisher count**: The number of nodes currently publishing to this topic.
* **Subscriber count**: The number of nodes currently subscribed to this topic.

Q10&Q11:



These commands display the **message structure** of Twist, and **subscribes to the /turtle1/cmd\_vel topic** and prints all messages sent to this topic in real-time.

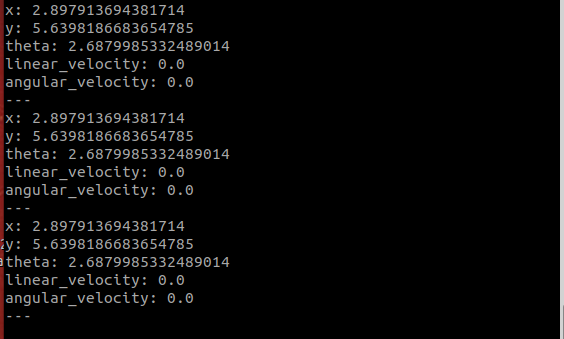
A black background with a black square

Description automatically generated with medium confidence

Q12:

A screen shot of a computer

Description automatically generated



A screen shot of a computer

Description automatically generated

The average hz is 62.5

Q13:

A computer screen shot of a virus

Description automatically generated

It draws a curve while I add a 0.5 angular speed on x axis.

A screenshot of a computer

Description automatically generated

Use rate of 3, it continuously drew a circle

Q14:

A screenshot of a computer screen

Description automatically generated

Add -t option to ros2 service list

A computer screen with white text

Description automatically generated

We then can see the type of the services.

The meaning of service type can be checked by using: ros2 interface show std\_srvs/srv/Empty

This will output:

---

This means that:

The Empty service **does not take any request parameters**.

The response is also **empty**, meaning no data is returned.

Calling the /clear service **only triggers an action** (e.g., clearing the turtlesim screen), but **does not require input parameters**.

The response does **not** contain any data, so it only confirms that the request was received.

Q15:

A screenshot of a computer screen

Description automatically generated

The turtlesim has been cleared

Challenge:

A screenshot of a computer

Description automatically generated

Use interface show, we know there is 5 parameters for set\_pen, they are “r” “g” “b” “width” and “off”

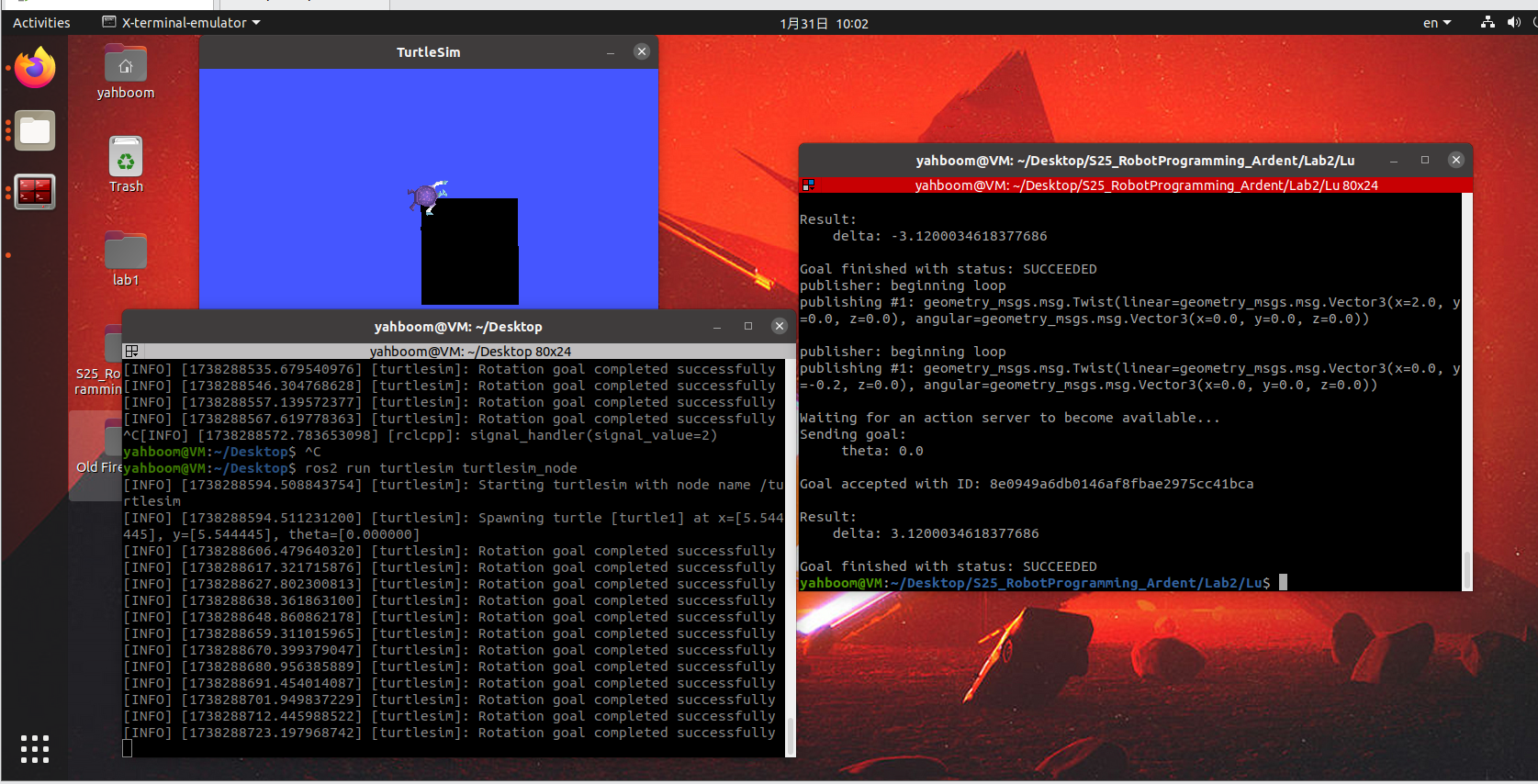
By setting r to 255, we get a red line.

Q17 & 18:

A screenshot of a computer

Description automatically generated

## Task3



A screenshot of a computer

Description automatically generated