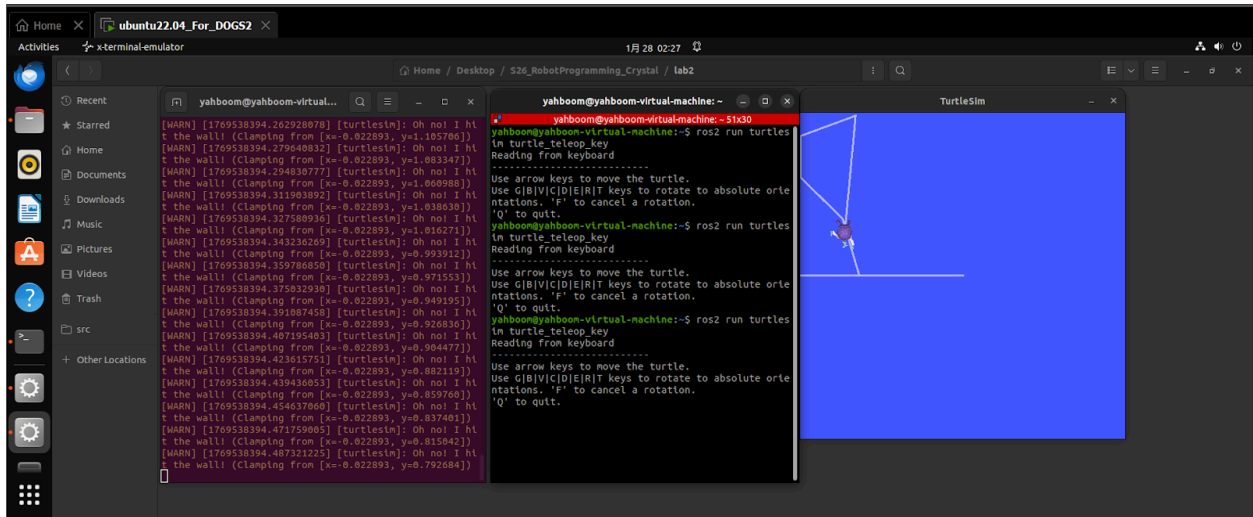


Task 2:

Starting Nodes

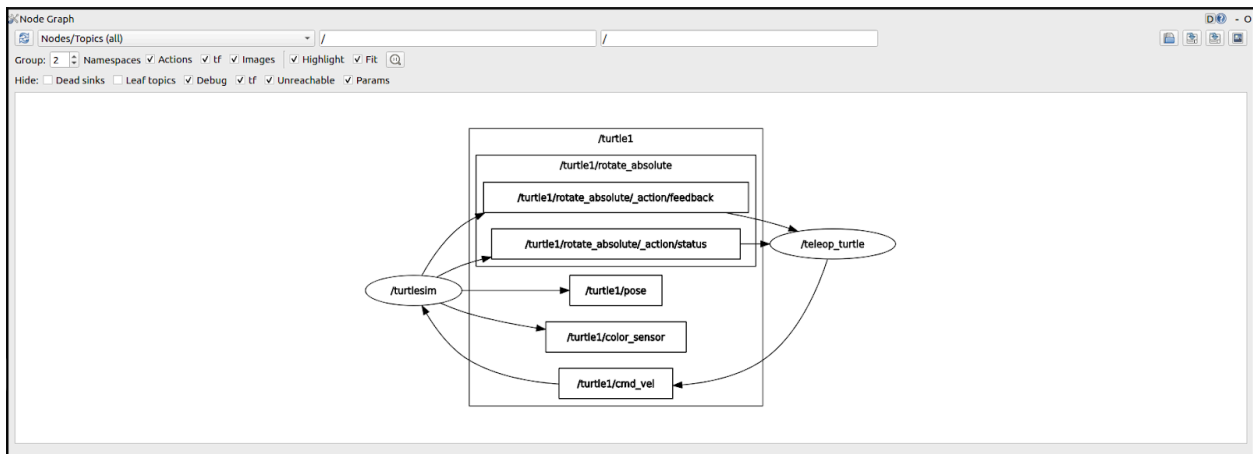


Screenshot of three windows open at the same time

Exploring Components

Topics

Step 7:



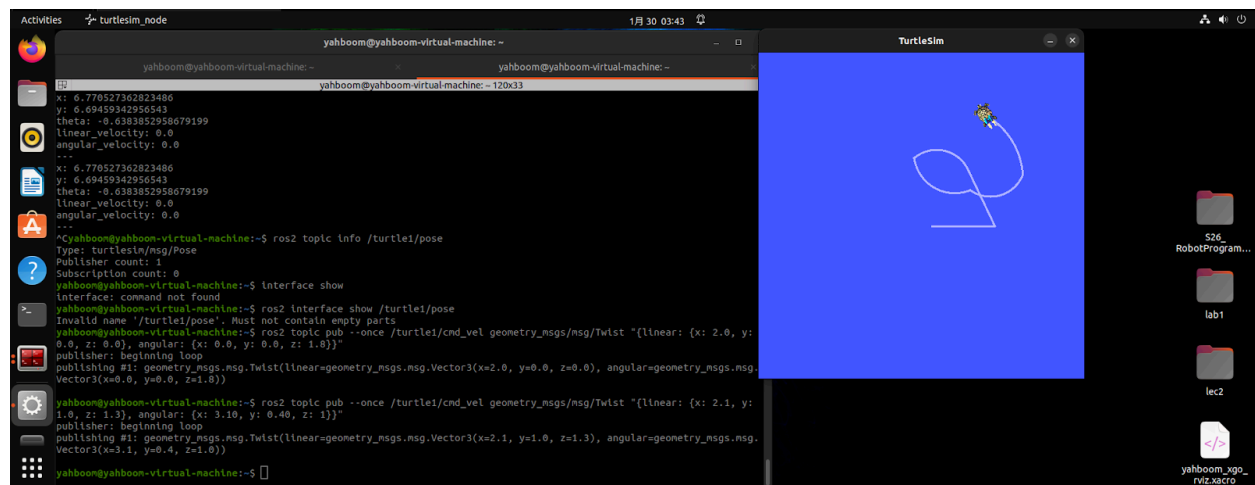
Flow diagram of turtle sim

Step 21:

```
yahboom@yahboom-virtual-machine: ~  
yahboom@yahboom-virtual-machine: ~  
yahboom@yahboom-virtual-machine: ~ 120x33  
linear_velocity: 0.0  
angular_velocity: 0.0  
---  
x: 7.255010604858398  
y: 9.085269927978516  
theta: 2.1840147972106934  
linear_velocity: 0.0  
angular_velocity: 0.0  
---  
x: 7.255010604858398  
y: 9.085269927978516  
theta: 2.1840147972106934  
linear_velocity: 0.0  
angular_velocity: 0.0  
---  
x: 7.255010604858398  
y: 9.085269927978516  
theta: 2.1840147972106934  
linear_velocity: 0.0  
angular_velocity: 0.0  
---  
^Cyahboom@yahboom-virtual-machine:~$ ros2 topic info /turtle1/pose  
Type: turtlesim/msg/Pose  
Publisher count: 1  
Subscription count: 0  
yahboom@yahboom-virtual-machine:~$ ros2 interface show turtlesim/msg/Pose  
float32 x  
float32 y  
float32 theta  
  
float32 linear_velocity  
float32 angular_velocity
```

Terminal outputs for three command lines

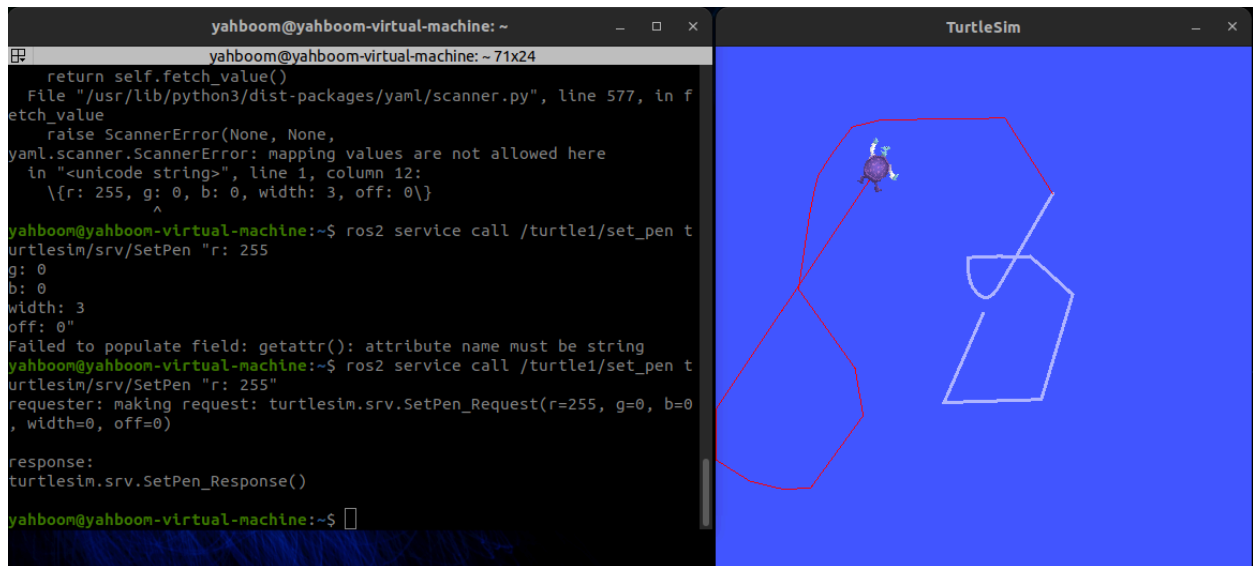
Step 25:



Screenshot of direct terminal control of the turtle

Services

Challenge:

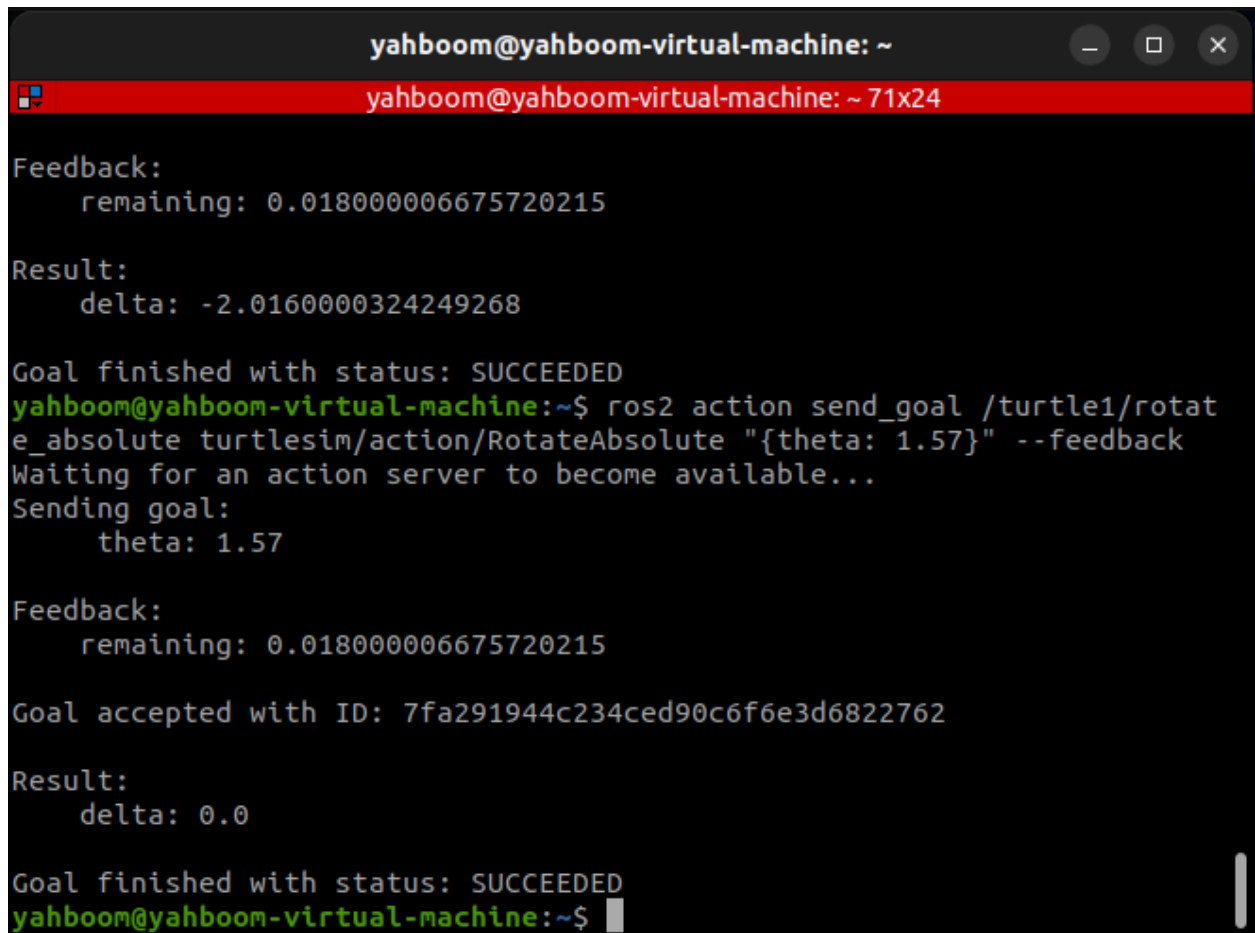


```
yahboom@yahboom-virtual-machine: ~  
yahboom@yahboom-virtual-machine: ~ 71x24  
return self.fetch_value()  
File "/usr/lib/python3/dist-packages/yaml/scanner.py", line 577, in f  
etch_value  
    raise ScannerError(None, None,  
yaml.scanner.ScannerError: mapping values are not allowed here  
  in "<unicode string>", line 1, column 12:  
    \{r: 255, g: 0, b: 0, width: 3, off: 0\}  
    ^  
yahboom@yahboom-virtual-machine:~$ ros2 service call /turtle1/set_pen t  
urtlesim/srv/SetPen "r: 255  
g: 0  
b: 0  
width: 3  
off: 0"  
Failed to populate field: getattr(): attribute name must be string  
yahboom@yahboom-virtual-machine:~$ ros2 service call /turtle1/set_pen t  
urtlesim/srv/SetPen "r: 255"  
requester: making request: turtlesim.srv.SetPen_Request(r=255, g=0, b=0  
, width=0, off=0)  
response:  
turtlesim.srv.SetPen_Response()  
yahboom@yahboom-virtual-machine:~$
```

Screenshot of the command to turn the line for the turtle's path red, and the subsequent red path

Action

Step 8:



```
yahboom@yahboom-virtual-machine: ~  
yahboom@yahboom-virtual-machine: ~ 71x24  
  
Feedback:  
  remaining: 0.018000006675720215  
  
Result:  
  delta: -2.0160000324249268  
  
Goal finished with status: SUCCEEDED  
yahboom@yahboom-virtual-machine:~$ ros2 action send_goal /turtle1/rotate_absolute turtlesim/action/RotateAbsolute "{theta: 1.57}" --feedback  
Waiting for an action server to become available...  
Sending goal:  
  theta: 1.57  
  
Feedback:  
  remaining: 0.018000006675720215  
  
Goal accepted with ID: 7fa291944c234ced90c6f6e3d6822762  
  
Result:  
  delta: 0.0  
  
Goal finished with status: SUCCEEDED  
yahboom@yahboom-virtual-machine:~$
```

Screenshot of terminal output with feedback

Task 3:

```

yahboom@yahboom-virtual-machine: ~
Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=wayland to run on Wayland anyway.
[INFO] [1769976459.889698835] [turtlesim]: Starting turtlesim with node name /turtlesim
[INFO] [1769976459.815244378] [turtlesim]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]
[INFO] [1769976487.795879336] [turtlesim]: Rotation goal completed successfully
[INFO] [1769976501.634414193] [turtlesim]: Rotation goal completed successfully
[INFO] [1769976525.316061218] [turtlesim]: Rotation goal completed successfully
[INFO] [1769976575.058810718] [turtlesim]: Rotation goal completed successfully

yahboom@yahboom-virtual-machine: ~
yahboom@yahboom-virtual-machine: ~$ 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.

yahboom@yahboom-virtual-machine: ~$ ros2 action send_goal /turtle1/rotate_absolute turtlesim/action/RotateAbsolute "{theta: 1.57}"
Waiting for an action server to become available...
Sending goal:
  theta: 1.57
Goal accepted with ID: 6e76c6afeb7544cf8159eb509a54ab17
Result:
  delta: 0.0
Goal finished with status: SUCCEEDED
yahboom@yahboom-virtual-machine: ~$ ros2 action send_goal /turtle1/rotate_absolute turtlesim/action/RotateAbsolute "{theta: 3.14}"
Waiting for an action server to become available...
Sending goal:
  theta: 3.14
Goal accepted with ID: dca7c59ad28b49eab240546217f94dbd
Result:
  delta: -1.5839999914169312
Goal finished with status: SUCCEEDED
yahboom@yahboom-virtual-machine: ~$ ros2 action send_goal /turtle1/rotate_absolute turtlesim/action/RotateAbsolute "{theta: 4.71}"
Waiting for an action server to become available...
Sending goal:
  theta: 4.71
Goal accepted with ID: 68e31ef2a6f24f3ab5d355b34d3b1a54
Result:
  delta: -1.568000078201294
Goal finished with status: SUCCEEDED
yahboom@yahboom-virtual-machine: ~$

```

Command	Use
<code>ros2 service call /turtle1/set_pen turtlesim/srv/SetPen "{r: 0, g: 0, b: 0, width: 3}"</code>	Make line black
<code>ros2 action send_goal /turtle1/rotate_absolute turtlesim/action/RotateAbsolute "{theta: 1.57}"</code>	Turn turtle to 90 degrees
<code>ros2 action send_goal /turtle1/rotate_absolute turtlesim/action/RotateAbsolute "{theta: 3.14}"</code>	Turn turtle to 180 degrees
<code>ros2 action send_goal /turtle1/rotate_absolute turtlesim/action/RotateAbsolute "{theta: 4.71}"</code>	Turn turtle to 270 degrees
<code>ros2 run turtlesim turtle_teleop_key</code>	Allow use of arrow keys to make turn move forward in direction it is pointing

My method involved using the telop command and rotate absolute commands to create a square in the turtle sim. I started by setting the line to black as specified in the lab instructions, and set the line width to 3 to allow it to be more easily seen. I then moved to the terminal containing the telop command, and hit the up arrow to move the turtle forward by one length. I then went back to my second terminal and entered the command required to rotate the turtle to face 90 degrees, then went back to the telop terminal to move it one length forward again. I repeated this process for 180 and 270 degrees, which finished the square