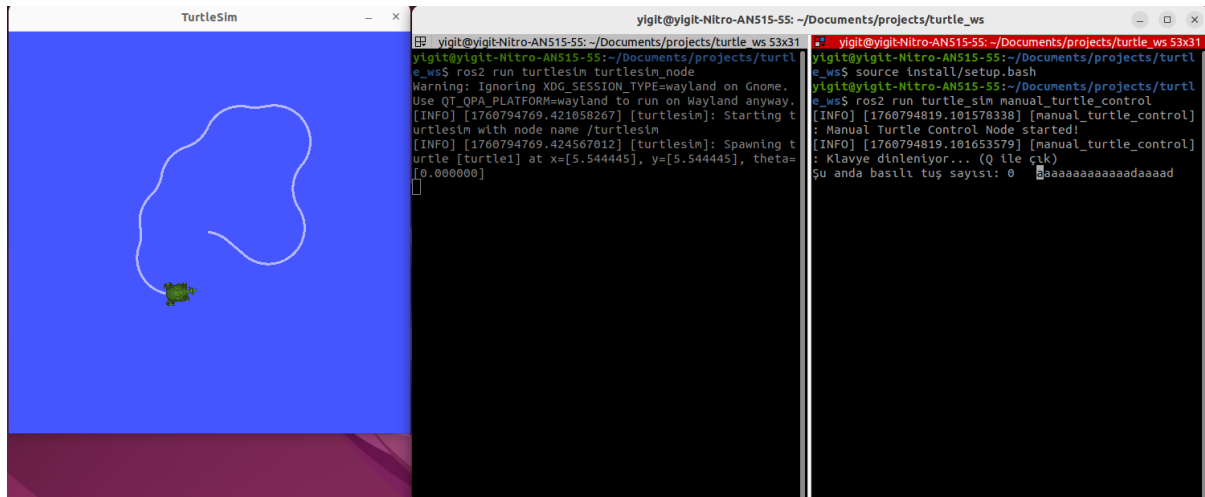


ITU ZES Solar Car Team Autonomous Group

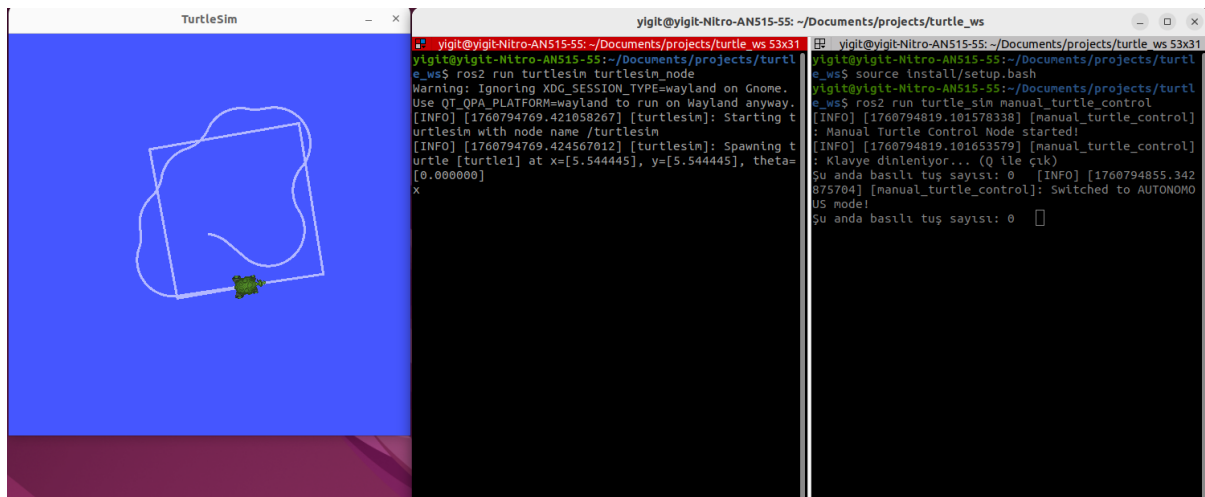
Homework 1

3 - Turtlesim and Keyboard Controller

Screenshot of your running node controlling the turtle manually:



Screenshot showing autonomous mode activated (with printed mode change).



Short explanation describing how you implemented mode switching:

When user pressed button “x”, `bool autonomous_mode = false`; turns to `bool autonomous_mode = true`; The node ignores keyboard input and instead automatically commands the turtle to move in a square path by publishing pre-defined movement and turn patterns.

Explanations and Observations for Each Section:

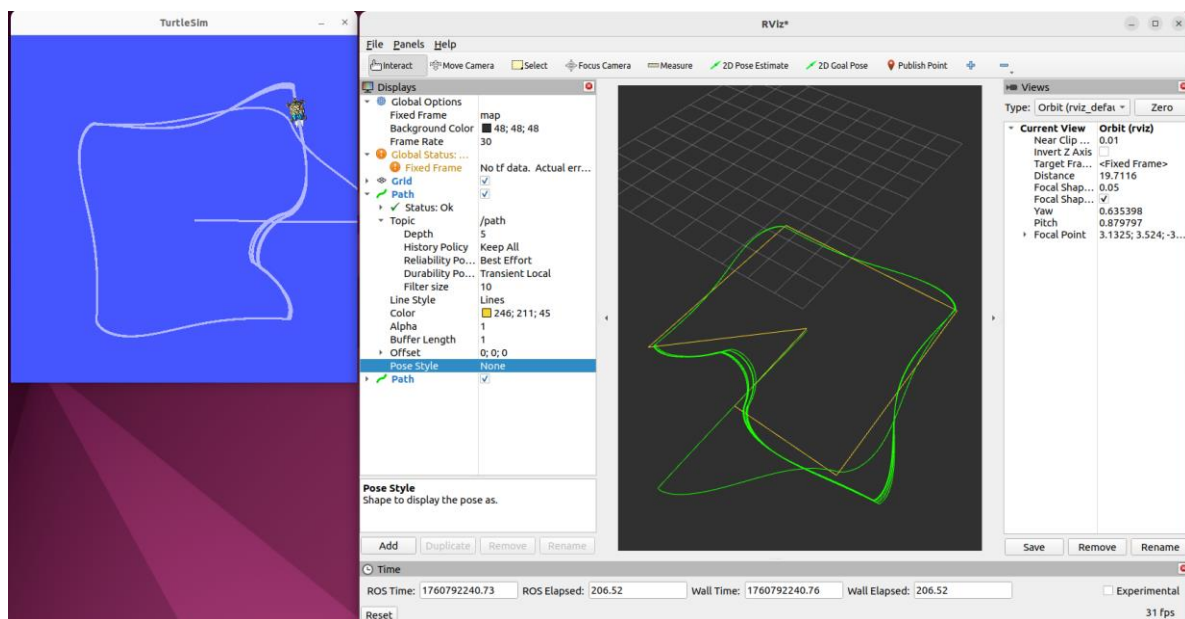
To be able to make turtle move smoothly when two buttons are pressed, a node listens how many buttons of keyboard is pressed.

Discussion of Challenges Faced and How You Solved Them:

Turtle was not moving smoothly when I press two buttons at same time. I added a node that tells me how many buttons are pressed. I use this number in if-else blocks to change how turtle behaves.

4 - Path Following with Stanley Controller

Screenshot of your visualization (path + trajectory) from rviz:



Terminal output showing total accumulated lateral error:

```

yigit@yigit-Nitro-AN515-55:~/Documents/projects/stan2_ws$ ros2 run t
urtle_path_following stanley_controller
[INFO] [1760792278.925605539] [stanley_controller]: ✓ Path received
with 7 waypoints
[INFO] [1760792278.946554509] [stanley_controller]: [Stanley] v=1.50
|  $\delta$ =0.03 |  $e_y$ =-0.132 |  $\theta_e$ =0.163 | TotalErr=0.007 | SegIdx=2
[INFO] [1760792278.996627067] [stanley_controller]: [Stanley] v=1.50
|  $\delta$ =0.04 |  $e_y$ =-0.120 |  $\theta_e$ =0.162 | TotalErr=0.013 | SegIdx=2
[INFO] [1760792279.046614642] [stanley_controller]: [Stanley] v=1.45
|  $\delta$ =0.05 |  $e_y$ =-0.109 |  $\theta_e$ =0.160 | TotalErr=0.018 | SegIdx=2
[INFO] [1760792279.096619331] [stanley_controller]: [Stanley] v=1.44
|  $\delta$ =0.06 |  $e_y$ =-0.098 |  $\theta_e$ =0.157 | TotalErr=0.023 | SegIdx=2
[INFO] [1760792279.146599566] [stanley_controller]: [Stanley] v=1.44
|  $\delta$ =0.07 |  $e_y$ =-0.087 |  $\theta_e$ =0.154 | TotalErr=0.027 | SegIdx=2
[INFO] [1760792279.196519310] [stanley_controller]: [Stanley] v=1.43
|  $\delta$ =0.07 |  $e_y$ =-0.077 |  $\theta_e$ =0.151 | TotalErr=0.031 | SegIdx=2
[INFO] [1760792279.246527304] [stanley_controller]: [Stanley] v=1.42
|  $\delta$ =0.08 |  $e_y$ =-0.063 |  $\theta_e$ =0.146 | TotalErr=0.034 | SegIdx=2
[INFO] [1760792279.296487756] [stanley_controller]: [Stanley] v=1.42
|  $\delta$ =0.09 |  $e_y$ =-0.054 |  $\theta_e$ =0.142 | TotalErr=0.037 | SegIdx=2

```

Explanation of how you computed the error term:

The Stanley Controller combines cross-track error (how far the turtle is sideways from the path) and heading error (how much the turtle's direction differs from the path's direction) and makes turtle turn more if its far from the path (e_y) or facing the wrong way (θ_e).

Explanations and Observations for Each Section: Stanley Controller is a great way to minimize error.

Discussion of Challenges Faced and How You Solved Them: My error margin was too high. I changed parameters, increased waypoints to minimize error.