

Turtlesim Automata

Liam J. Nolan
MCI

The purpose of this assignment was to create a edge detection program for the ROS2 Turtlesim node. This node will detect when the turtle hits an edge, turn the turtle 90 degrees and continue forward. The node was written in C++ and a custom bash script will install the relevant files

Procedures Followed:

To initialize the turtle a function (teleport_turtle) was written. This function spawns the turtle in the center of the screen with a random theta and commands the turtle forward at a specified velocity.

Upon running the program a boolean flag check it completed to determine if it the first time the program was run. If it is the first time, the teleport turtle function will run and the flag state will be changed allowing the program to enter the edge detection function.

To detect edges, the program subscribes the turtle pose message. The program then continuously checks whether the turtle is within the bounds of the screen. If the the turtle reaches the bounds, the program publishes a theta velocity that rotates the turtle until 90 degrees is reached. The program then commands the turtle forward and this will run until infinity.

Challenges:

Due to the nature of the edge detection, the turtle will often detect an edge immediately after rotating, causing the turtle to rotate and additional 90 degrees. This could easily be fixed by adding a delay in the edge detection check however this was outside the scope of the assignment and as a result was not implemented.

Success:

Overall the program works as intended. The program successfully spawns the turtle in with a random angle and the turtle will change direction when an edge is detected. It was an excellent opportunity to become better versed in ROS2 and helped create an understanding of nodes.

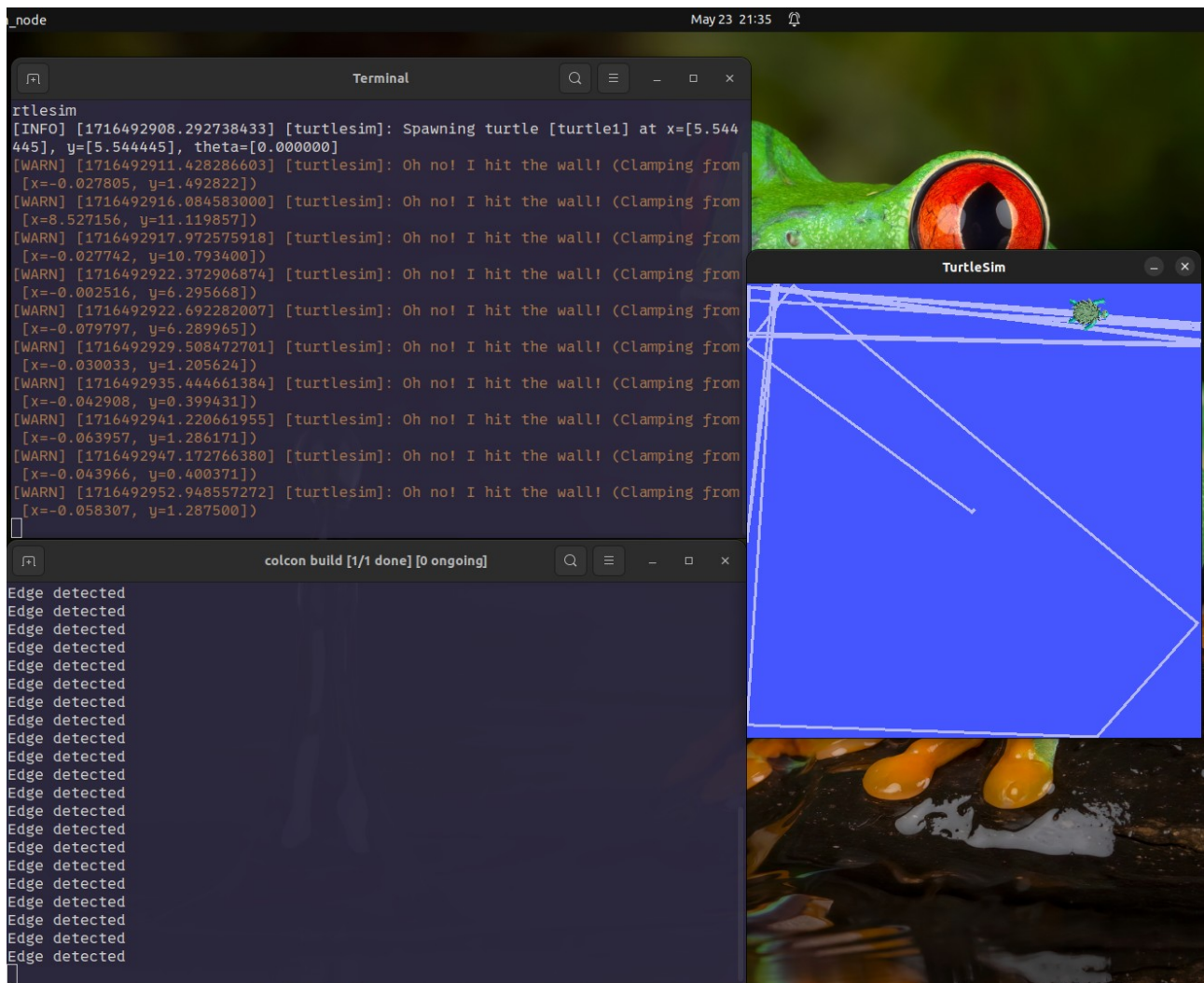


Fig. 1: Terminal and Turtlesim Output