CS460 Homework 1

Josh Hipps

Due 9/17/2025

For homework 1, I felt as if each of the ROS publishers we made were very similar in essence. The skeleton of each publisher, i.e. the turtle node class' general structure, was created in comparison with the code in Dr. Crawford's square publisher example

Beginning with rectangle, I found that a 5-unit line (5 periods of linear velocity = 1.0) was simple to create for my short edge. Next, I experimented with turtlesim trials to find the value that gave me something close to a 90-degree turn, which resulted in a 2 second turn with 1.58 z-angular-velocity. The rest of the shape was created by sequencing a 10 unit edge, a 90-degree turn, a 5 unit edge, a 90 degree turn, and finalized with a 10-unit edge. The long edges of the rectangle are double the distance of the short sides.

Second, I saw the diamond as nearly identical to the rectangle but with alternating angle sizes instead of edge lengths. I firstly reduced each straight-line drawing to 5 units. Then, I sequenced a 45-degree turn, a 135 degree turn, and another 45-degree turn, with 5-unit lines in between each. This created two obtuse angles and two acute angles, for a diamond that sits on the x-axis like a rhombus.

Third, I realized that the circle would be easily created by a constant linear and angular velocity. I calculated linear and angular velocity with the relationship that radius = linear/angular velocity, and implemented a radius of 2 units for the circle. After 32 increments of the timer, the large circle completes itself and movement stops.

For the random publisher, I wanted to do something that honed the basic skillset I had gained on the first three publishers. I chose to create the Steam logo, one of the most frequently used applications on my computer. I began with a 2-unit radius circle, moved the turtle inward on the circle and created a .5-unit radius circle, and then created a second 0.8-unit radius circle to represent the larger gear in the logo.

I noticed that the output in turtlesim was inconsistent for every single publisher, meaning that it varied from run to run even with the same code, but I could not figure out how to avoid that issue and it may have caused inaccuracy.