Eland Anthony

CS 460-001

WanderBot

My python script for my wanderbot sets linear speed to 1.0 and angular speed to 0.6. This is a relatively high movement speed and allows my robot to reach the green zone in 4 seconds starting from the yellow zone. Conversely, starting in the green zone the bot can reach the yellow zone in 6 seconds, though this can take much longer as there are more openings for the robot to exploit that are away from the green zone. Reaching the red zone will take the longest of the aforementioned moves. This is due to the small opening as well as the long narrow path it upholds which causes more occurrences of change of movement than normal – this increases the changes of my ‘rotate 180 degrees’ code segment. The fastest time I was able to record for reaching the red zone was from the green zone at about 25 seconds.

My overall strategy for this project was a simple reactive decision logic. I made the logic as simple as possible. This strategy succeeded, although it was difficult to gauge what the linear and angular velocities and other metrics in my code should be. This was mostly perfected by ‘guess and check.’ One strategic point I missed early on was I did not realize that regardless of how ‘perfect’ I made the robot react, it would still get stuck in corners as well as in the red zone tunnel. In order to change this while staying with a reactive approach, I decided to randomly issue ‘rotate 180 degrees’ commands (occurs in about 1/100 moves). This only added to the complexity of the code and I spent further time deciphering what I needed to write to uphold my speed and performance as well as random perfect 180 degree turns without crashing. I am very satisfied with my robot’s performance and do not plan on adding anything significant to the script; I loved the project once I was finally setup. I ended up using Lubuntu in VM – I just couldn’t ever get suited to RDS.