Lauren Paredes lapa5112@colorado.edu
Nova White Nalan.Yilmaz@colorado.edu
Ali Keller alke5276@colorado.edu

Project Name: A Day at the Park

Abstract

A day at the park features many different types of robots enjoying the weather. They are at a park which we created where humans are not allowed. Inside the park you will see the TIAGo steel 'moms' following along the shortest path around the pond. In the pond you will see turtle bot 'ducks' swimming around the pond staying in a group. Then in the playground there are multiple E-pucks 'kids' running around while avoiding obstacles.

Equipment

1. Webots
2. Tiago steel
3. Turtle bot
4. E-puck

Deliverables and Implementation Plan

- Create our world in webots Lead--Ali
  - This was the first iteration of the map that includes things from the list below. In earlier steps of the lab the map was at an unknown offset that caused us many issues. We eventually gave our world a translation that made it so we no longer need an offset which fixed many problems.
  - Creating obstacles
    - Rocks
    - Pond
    - Benches
    - Trees
    - Fence
    - Other robots
- Implement code TIAGo steel Lead -- All
  - Using lab 5 as a template we started out trying to get the tiago to follow a short path. However we were encountering many problems with this. As our original world was not all positive within webots. So then we all tried adding 15 to each spot the pose was calculated to try to counteract the negative parts of the map. However once we got that we encountered two more major issues: A* path planner took a very long time to compute and the robot would not always follow the path. We then realized-after countless hours of struggling to fix it-the problem was the map offset. When we fixed that the TIAGos were able to be placed around the map, path plan, and then follow the path. However the A*'s path planner still

took a very long time. To help this we took away free space so that there were less calculations. Now the TIAGos make the shortest path around the pond.

- ➢ Add TIAGos
  - ■ Get outline from lab 5
    - Use the path_planner A* code and the state machine outline
      - ◆ Manually load the map and save as a npy file
        - ➢ Drive the TIAGo around the world so it can map all the obstacles and free space
          - ■ Displays the mapping on a small black screen in real time
      - ◆ Use planner mode to find a path
        - ➢ Start and end points need to be converted into world frame coordinates
        - ➢ Convolve the map from the LIDAR readings --- must make the kernel larger so that the borders are larger so there is less computing time
        - ➢ Save the path as a npy file
      - ◆ Use the autonomous mode to follow path
        - ➢ Load in saved path
        - ➢ Go through and calculate rho and alpha errors and use them to help move along path

- ❖ Implement code Epucks Lead -- Nova
  - ➢ At first we decided to not use the E-pucks since they were too small to be spotted in our world. However, as the abstract of our project changed we thought E-pucks would be a valuable addition to the family. We began by hoping to have them follow a yellow line located on the road that was in our previous map, as if they were crossing the street. Once the map changed, we decided to alter the implementation to take place within the playground. We thought the most productive way to do this was to program the E-pucks in a way where they avoided the obstacles placed inside the playground and just roamed around their limited area.
  - ➢ Add the Epucks
    - ■ 8 E-pucks were placed randomly inside the playground
      - They follow their given path until an obstacle comes their way
      - Obstacles are detected through the built in ground sensors
      - Speed is set in a way where it prevents excessive turning of the pucks
      - Sensing is set to be 100ft so they can have more wiggle room around the playground.

- Implementing code for Turtle bots Lead-- Lauren
  - The turtle bots were originally supposed to be the ducklings of the Tiago's but we realized that it was much harder to get the path following to work. Instead of using a predetermined path we decided that if the turtle bots could just go around in the pond it would be better. They now use how long and far(calculated from the position sensor) they traveled to go around the pond making a large polygon. They have different offsets so that they do not crash into each other.
  - Add turtle bots
    - Get one turtle to make a polygon around the pond
      - This needs to have a long enough side length to get around while maintaining a long enough turn duration to actually turn
      - Need to use position sensor to be able to calculate distance traveled
      - Need to use the time step provided by webots to calculate turning time
    - Make side lengths shorter
      - Must be within the pond
    - Add more Turtle bots
      - Make sure that they have enough offset to not crash into one another.