EEP 545 Final Project
Due date: Monday December 10

Welcome to the final project of the course! We hope you've enjoyed it, we certainly have! You will now combine all of your past blood, sweat, tears, and knowledge from the course to autonomously navigate your robot as quickly as possible.

1 Task

You will be given a series of waypoints located in a map, as illustrated in Fig. 1:

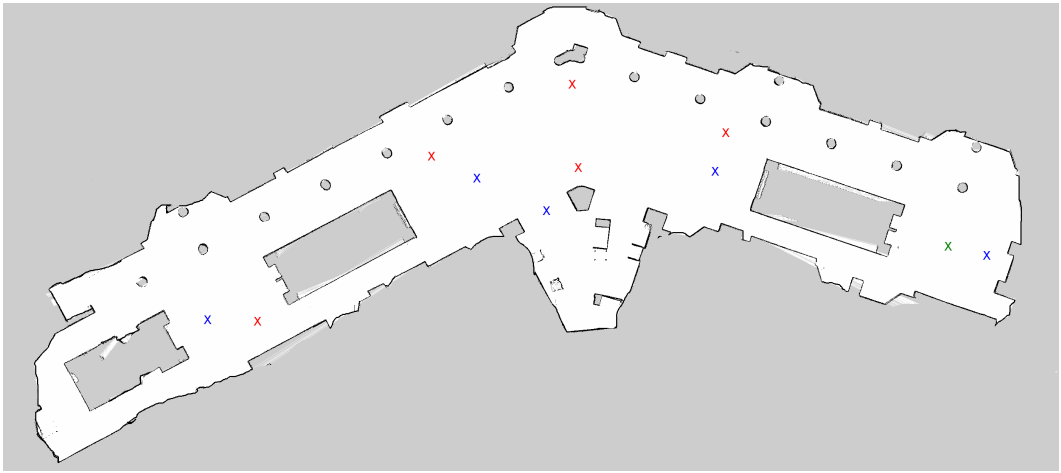

Figure 1: Map with marked waypoints in blue and red. The car should traverse over blue waypoints, and not traverse over red waypoints. The green x denotes where the robot will start.

You will receive three csv files - one containing waypoints that the robot should traverse, one containing waypoints that the robot should not traverse, and one containing the starting point of the robot. Each row of the csv files represents a single (x,y) waypoint **in pixels** (recall that many of the labs have a function inside of *utils.py* called *map_to_world()* that will convert pixel coordinates to world coordinates). Theta will not be specified. Also, the robot does not need to visit the desired waypoints in any particular order. Each of the waypoints will also be physically marked in the world with colored paper - either with blue paper that should be driven over, or red paper that should not be driven over. Note that the robot has only visited a waypoint if it actually drives over any part of the paper - just getting close does not count.

The requirements for the simulation-only track are slightly different; please see the final section of this document.

2 Scoring

You will gain points for doing the following:

1. Driving over blue waypoints (+1 each)
2. Visiting all of the blue waypoints quicker than a given baseline amount of time (8 minutes) without running over any red waypoints (+3)

3. Being one of the three groups to visit all blue waypoints quickest (1st place -> +4, 2nd place -> +3, 3rd place -> +2). Note that there is no requirement that you don't run over any red waypoints here.
4. Implementing something particularly creative or awesome (Case-by-case basis)

You will lose points for doing the following:

1. Driving over red waypoints (-1 each)
2. Damaging your robot (Case-by-case basis)

3 Submission

You will also need to submit a report describing your system and the design choices that you made in developing it. It should be written and sufficiently detailed such that any class member outside of your group could re-implement your system.

4 Additional Notes

- Each team will receive a block of twelve minutes for their demo.
- The waypoints will be placed in the world ***only approximately*** at the locations specified in the csv files. Therefore, it may be a good idea to use computer vision to detect waypoints once your robot comes close to them.
- You are allowed to restart your run as many times as you would like within your allotted time slot. Each time that you restart, your robot will be returned to the starting point and your score will be reverted to zero. Your final score will be the maximum score achieved across all of your runs.
- You can also intervene in the middle of a run. This involves anything that interferes with the robot's autonomous behavior, such as picking up the robot or using RVIZ to re-localize the robot. However each intervention will result in the loss of one point for the run. The clock will keep ticking during your interventions.
- Each group will nominate a team captain. Although other group members can voice their opinions, the team captain will have the final say on if/when/how interventions and restarts take place
- You are allowed to change code/settings during your demo period. This will also count as part of the demo time.
- A run ends either when you have visited all blue waypoints, or you can declare it to be over at any time. At that point, you can start another run if you still have time remaining.
- As previously mentioned, a waypoint (good or bad) will be considered visited if any part of the robot physically touches any part of the waypoint.

5 Simulation Only

Due to the logistical difficulty of determining if a simulated robot has touched a waypoint, the simulation only track will not be required to reach specific waypoints. Instead, the robot needs to navigate to two areas on the map, as shown in Fig. 2.

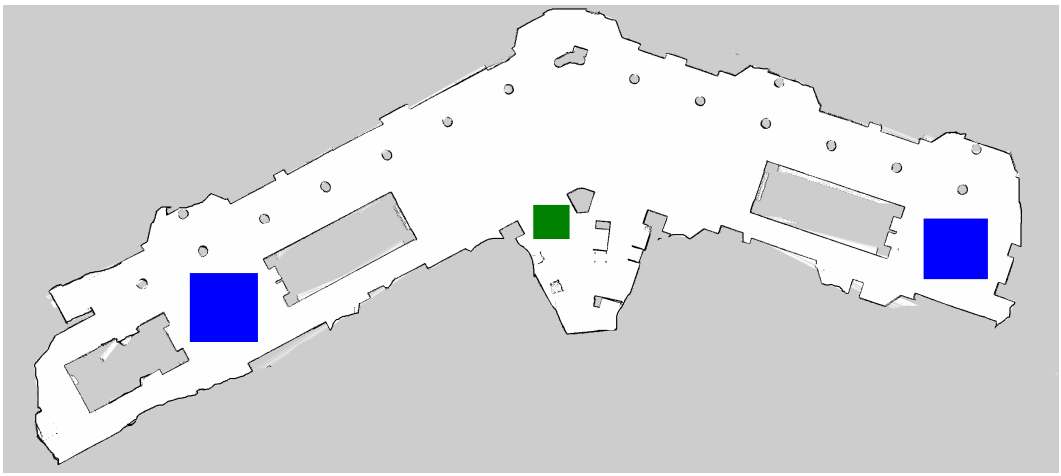

Figure 2: The blue rectangles denote areas that the robot must reach. The green square indicates the area at which the robot should start.

To receive full credit:

1. The robot should autonomously visit both of the indicated areas in 5 minutes or less.
2. Do not use any of the outputs from the *sim_car_pose* node. In other words, **do not subscribe to */sim_car_pose/pose***, nor lookup the transform between the */map* coordinate frame and the *sim_pose* coordinate frame.

Please submit the following:

1. All of your code
2. A video that demonstrates your system
3. A bag file (include all topics) that demonstrates your system