

## Assignment-1

*Released: January 27<sup>th</sup>, 2021**Deadline: Feb. 15<sup>th</sup>, 2021*

## Instructions

- Submit your code files and a report (or everything as a single jupyter notebook if you wish) as a zip file, named as  $\langle \text{TeamName} \rangle .\text{zip}$ .
- The report should briefly describe your approach/algorithm used, your final results and any other findings.
- The deadline is Feb 15th, 23:55. This is a hard deadline and no submissions will be accepted after this.
- Plagiarism detectors will be run on all submissions, so please do not copy. If found, you would be given a straight zero for the assignment.

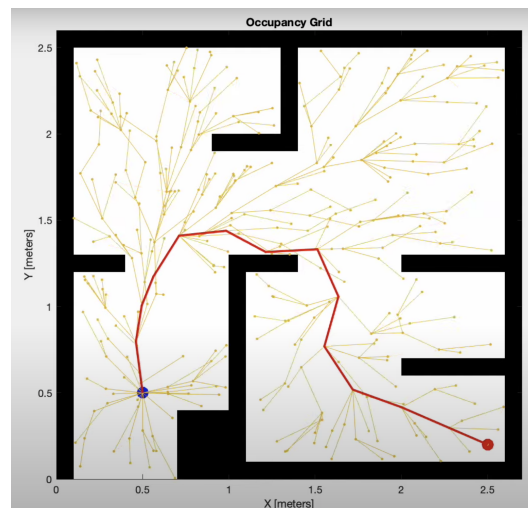
Welcome to the first assignment of the course! You are living on the desert planet of Tatooine, and you have been tasked with building a robot for the Resistance. You have already put together all the circuits, but you still have to solve the robotic path planning problem. You need to help the robot navigate from its initial location to its destination, whilst efficiently avoiding stormtrooper obstacles on the way.

## Rapidly-exploring Random Tree (RRT)

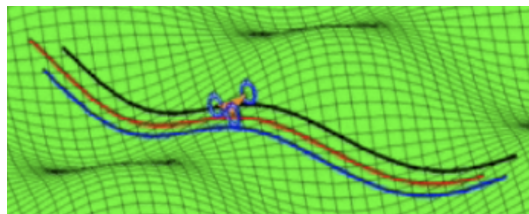
The goal of this assignment is to implement the RRT path planning algorithm for a robot. The robot is to navigate a two dimensional space, avoiding known locations with obstacles, traveling from its initial location to a goal location. Given localization information (robot's initial position, obstacle location, goal location), your task is to implement a path planning decision maker to drive the robot from its initial position to the desired location.

Specifically, the problem can be formed as follows: Consider a 2D grid instantiated with different kinds of obstacles (for instance, geometrical shapes like Rectangles, Circles, Triangles or a combination of any of the above 2/3). Assign a start and end point on this grid. You must implement the RRT algorithm for two cases, (1) **Holonomic Robot** and (2) **Non-Holonomic Robot**.

You must submit **two videos** for each of the above cases, showing the evolution of your graphs. The first one should showcase only the platform center velocity, that is, the trajectory of the center of the robot. Something like this:



Your second video should show the **individual wheel trajectories** (evolution of all the wheels in all the possible paths). Ensure that no collisions happen near the boundaries and use methods explained in the class for this. Also, the trajectories for each individual wheel must appear consistent. The left wheel should not go haywire while the right wheel's trajectory is smooth - they must be in sync with one another. If you code the algorithm right, the kinematics will guarantee this! :)



You can refer to [this video](#), which shows the trajectories plotted for all three wheels of the robot. Although this is done in 3D, and you need to implement this for only two dimensions.

Hence, there will be 4 videos in total: 2 for holonomic and 2 more for non-holonomic. For each case, one video will show platform center velocity and the other, the individual wheel trajectories. You should save all the graphs, combine them into a video and upload it to your OneDrive or Google Drive, and include that link in your report. You should also include a few pictures of the graphs in your report, along with the video links. For generating videos from individual graphs, you may use [this script](#).

The more complex the grid, the more appreciation you get! At the same time make sure you don't fill the entire grid with obstacles. The obstacles can be made complex by combining different shapes like circles, rectangles or any other geometric shapes you prefer. Also, **fix the size of your robot and increase the size of the obstacles accordingly near the boundaries (as discussed in the class)**.

## Deliverables:

- Code for both the cases. You can use Python, MATLAB or whichever language you are comfortable with. Please ensure the code is well written, and we can ask you to explain certain snippets of it during the evaluations.
- Two graphs as explained above, for each case.
- A report summarising your understanding of the algorithms and explaining the results.
- Put all these files in a folder <TeamName>, zip it, and submit it on moodle.

Feel free to reach out to the TA's for any queries. All the best, and may the Force be with you.