EECS 195: Autonomous Systems - Spring 2020
Assignment #2
Due Date: 5/10/2020

Problem 1: Mapping using Hector SLAM - manual control (30 points)

In this problem, we are going to use TurtleBot3 simulator to build a map for the environment. We are going to manually move the robot until we build the map. First, we need to install the Hector SLAM package as follows:

```
>> sudo apt-get install ros-melodic-hector-mapping
```

Now, in three different terminals, launch the following launch files:

```
>> roslaunch turtlebot3_gazebo turtlebot3_stage_4.launch
>> roslaunch turtlebot3_slam turtlebot3_slam.launch
slam_methods:=hector
>> roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

Using the keyboard to move the robot around the environment. Observe the map that is being updated by the Hector SLAM node. Continue to drive the robot manually until the whole map is built. Once the map is ready, save the map using the map_server node as follows.

```
>> rosrun map_server map_saver -f ~/map
```

Submit the obtained map file.

Evaluation:
The grader is going to open your map file and check it for completeness and correctness.

Problem 2: Automatic mapping using Hector SLAM - automatic control (70 points)

In this problem, you are required to write a node called `build_map_automatic` that drives the robot in the environment to build its map. The node shall subscribe to the following topics:

/scan: this topic contains the information from the LiDAR scanner

/slam_out_pose: this topic contains the "estimated" pose of the robot from the localization algorithm

/map: this topic contains the occupancy map estimated by the SLAM algorithm

The node should then publish to the topic:

/cmd_vel: this topic is used to control the linear and the angular velocities of the robot.

The node should perform the following steps:
Step 1: Read the current occupancy map. Determine if more areas of the map need to be explored.
Step 2: Pick a point that is on the edge of what is currently explored. Move the robot to that point. Make sure the robot does not hit an obstacle while it moves to that point (you will need the LiDAR information to correct the course of the robot to avoid hitting the obstacles).
Step 3: Repeat steps 1 and 2 until the whole map is obtained. Save the map and printout the time used to build the map.

Note that the steps above are not complete. You will need to fill in the details and add any extra steps that are required to accomplish the task.

Evaluation:
The grader is going to evaluate your code in three different environments. These environments are unknown a priori. Make sure your code is general enough to work on any environment and from any initial pose. Grades are given based on (1) completeness and correctness of the obtained map and (2) time needed to build the map (the faster the better).