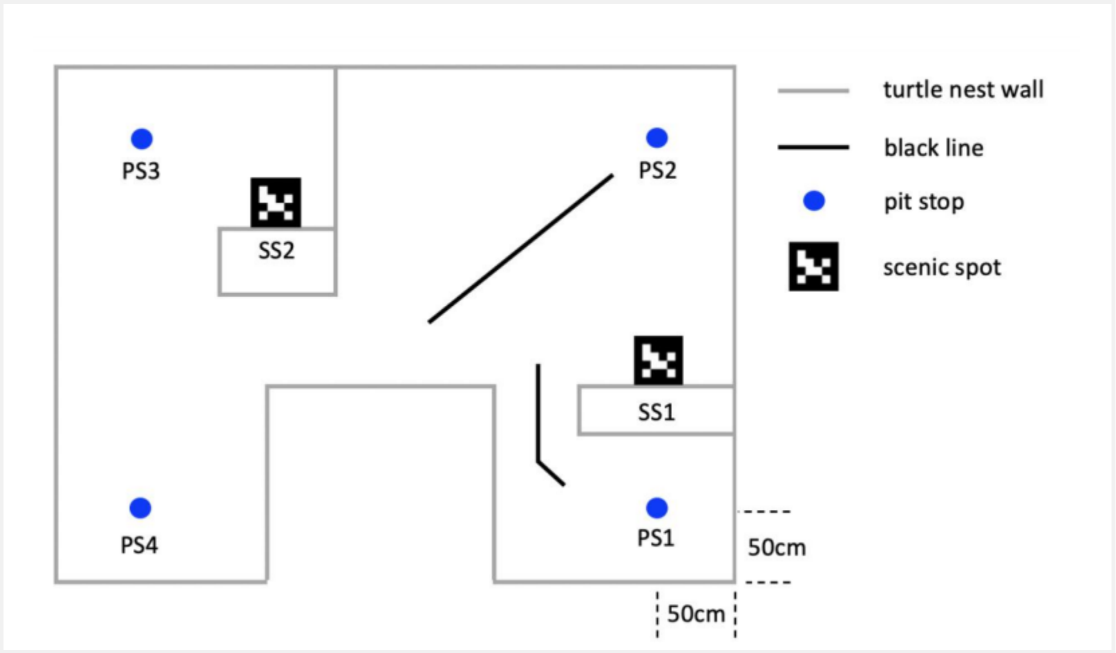
Capstone Report

Introduction

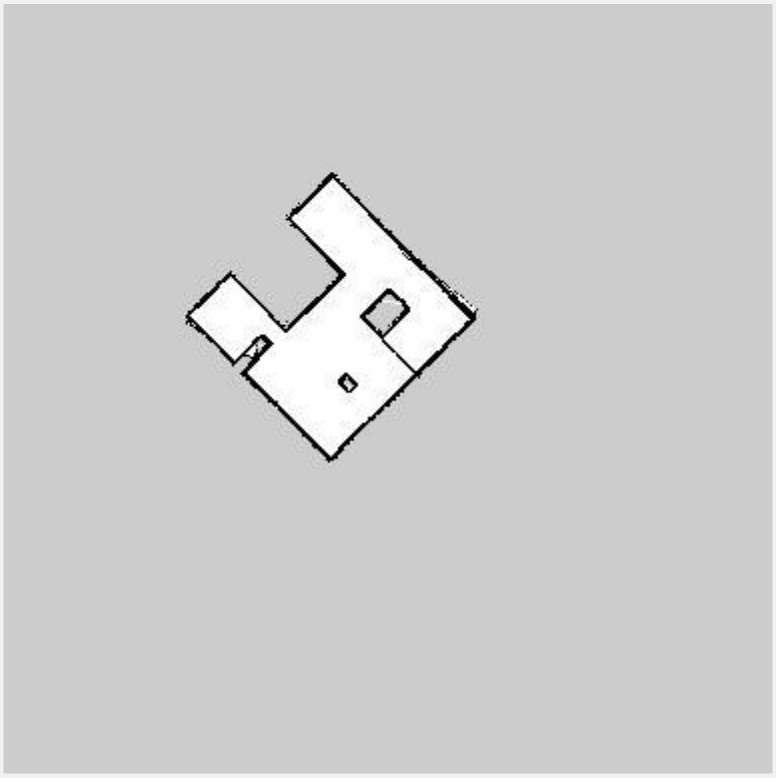
Robotic navigation can be implemented based on Adaptive Monte Carlo Localization.   We prepare a map before competition, which is generated from GMapping. Our goal of this competition is to make the robot move from the PS1 to PS2, PS3 and PS4 in order. In the process, the robot needs to figure the ar_tag and echo the correct sound.



Approach

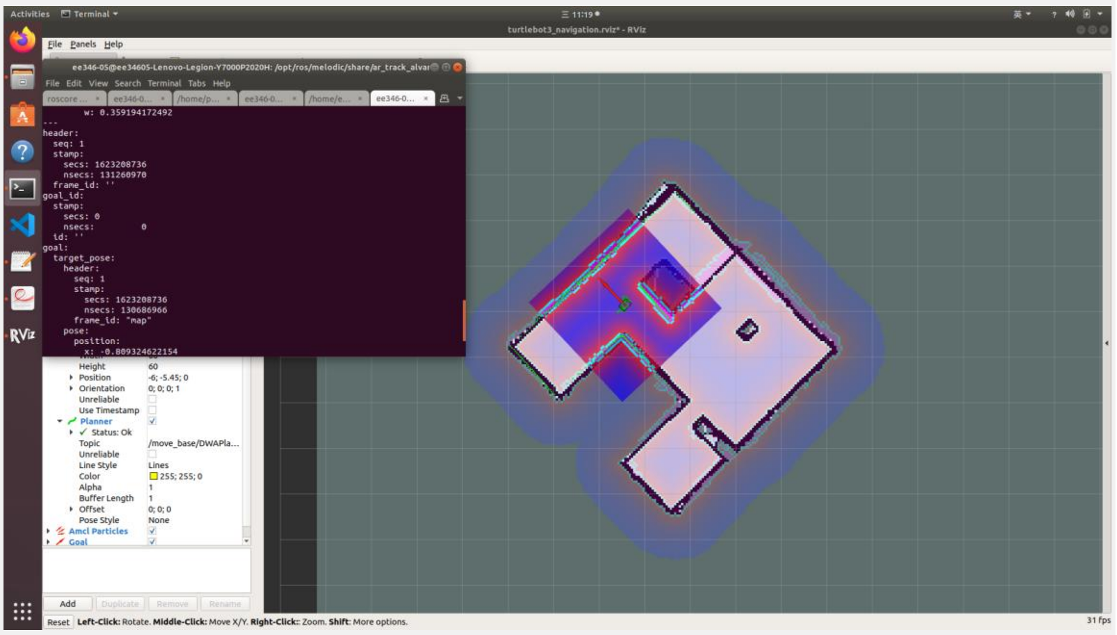
Mapping

We first built the map by controlling the robot using keyboard moving around the competition field.



Navigation

By using the tool of Rviz, we first set the initial pose and position corresponding to the map. Then we can set the goal which is the PS and SS in Rviz, so that we get the message of the goal’s pose and position from the rostopic echo method. Compare the real move and goal, we adjust the goal in Rviz and finally got the useful and correct goal’s pose and positon.



```
ee346-05@ee34605-Lenovo-Legion-Y7000P2020H: /opt/ros/melodic/share/ar_track_alvar
        w: 0.359194172492
---
header:
  seq: 1
  stamp:
    secs: 1623208736
    nsecs: 131260970
  frame_id: ''
goal_id:
  stamp:
    secs: 0
    nsecs:         0
  id: ''
goal:
  target_pose:
    header:
      seq: 1
      stamp:
        secs: 1623208736
        nsecs: 130686966
      frame_id: "map"
    pose:
      position:
        x: -0.809324622154
```

```
Height            60
Position          -6; -5.45; 0
Orientation       0; 0; 0; 1
Unreliable
Use Timestamp
Planner           ✓
  Status: Ok
  Topic           /move_base/DWAPla...
  Unreliable
  Line Style      Lines
  Color           255; 255; 0
  Alpha           1
  Buffer Length   1
  Offset          0; 0; 0
  Pose Style      None
Amcl Particles    ✓
Goal              ✓
```

Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click: Zoom. Shift: More options.

In the final, we set the following 7 goal to implement.

self.locations['one']   = Pose(Point(-3.43499898911,   1.24000000954, 0.000), Quaternion(0.000, 0.000,   -0.171919584032,   0.985110986959))#1

self.locations['one2']   = Pose(Point(-1.81499993801,   0.759999811649, 0.000), Quaternion(0.000, 0.000,   -0.32065647914, 0.94719555657))#0

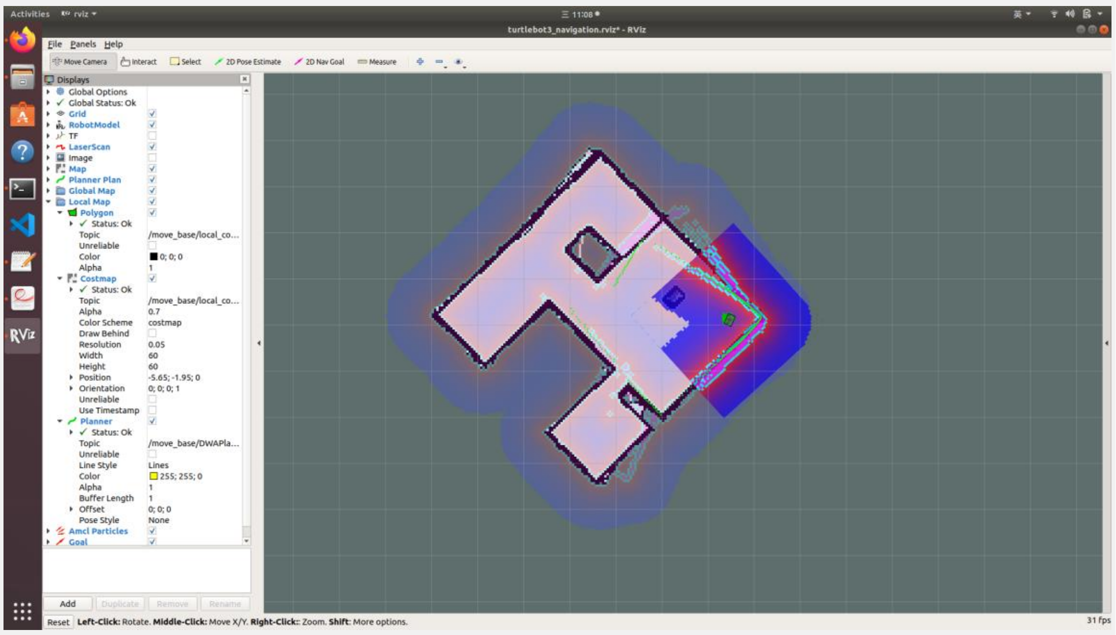
self.locations['two']   = Pose(Point(-1.88499975204,   -1.47999978065, 0.000), Quaternion(0.000, 0.000, 0.937552859467,   0.347842831902))#2

self.locations['three'] = Pose(Point(-3.04499912262,   -0.349999815226, 0.000), Quaternion(0.000, 0.000, 0.918319780663, 0.395839336655))#3

self.locations['four']   = Pose(Point(0.925000011921, 1.30999982357, 0.000), Quaternion(0.000, 0.000, 0.941795375126, -0.33618666153)) #4

self.locations['five']   = Pose(Point(0.212898686528, 0.743556082249, 0.000), Quaternion(0.000, 0.000, 0.928073755818, 0.372396433604)) #5

self.locations['six']   = Pose(Point(-1.94781160355, 4.23388195038, 0.000), Quaternion(0.000, 0.000, 0.922130849828, 0.386878140758)) #6



```
Global Options
✓ Global Status: Ok
Grid                  ✓
RobotModel            ✓
TF
LaserScan             ✓
Image
Map                   ✓
Planner Plan          ✓
Global Map            ✓
Local Map             ✓
  Polygon             ✓
    ✓ Status: Ok
    Topic             /move_base/local_co...
    Unreliable
    Color             0; 0; 0
    Alpha             1
  Costmap             ✓
    ✓ Status: Ok
    Topic             /move_base/local_co...
    Alpha             0.7
    Color Scheme      costmap
    Draw Behind
    Resolution        0.05
    Width             60
    Height            60
    Position          -5.65; -1.95; 0
    Orientation       0; 0; 0; 1
    Unreliable
    Use Timestamp
  Planner             ✓
    ✓ Status: Ok
    Topic             /move_base/DWAPla...
    Unreliable
    Line Style        Lines
    Color             255; 255; 0
    Alpha             1
    Buffer Length     1
    Offset            0; 0; 0
    Pose Style        None
Amcl Particles        ✓
Goal                  ✓
```

```
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click:: Zoom. Shift: More options.
```

Result and Conclusion

The accuracy of the map do influence the effect of this competition, and we find the robot in competition use a lot of time to correct the pose and position when it is very close to the goal. If we can optimize the parameters of the map and pose, or we can make a filter to help the robot finding its correct pose at the time, the final effect will be better.

Source code: https://github.com/LinZhanKun59/capstone