산학프로젝트 주간 보고서 ( 2019. 04. 18)

이름/팀명 : 장 환석/AP 프로젝트명 : AI planning

**1.개인별 주간 목표(상세 개발 계획서)**

|  |  |
| --- | --- |
| 6주차 | 1. A\* 알고리즘과 PDDL 이해 및 개발  2. 시뮬레이터 연동 |
| 7 주차 | 1. A\* 알고리즘과 PDDL 이해 및 개발  2. 시뮬레이터 연동 |

**2. 추진 실적**

**2-1 현재 진행 과정 Github repo에 등록**

<https://github.com/niklasjang/mybot_ws>

0. git clone

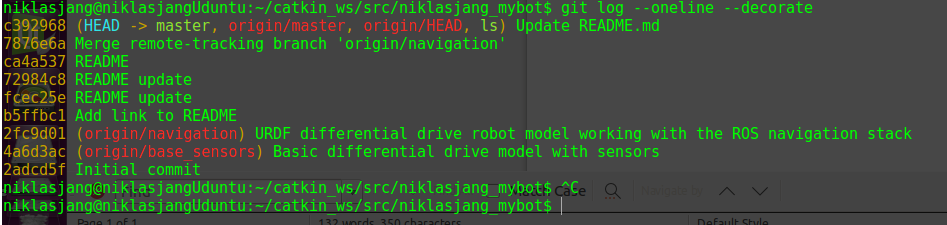
1. cd ~/catkin\_ws/src

2. git clone https://github.com/niklasjang/mybot\_ws.git niklasjang\_mybot

3. cd mybot\_ws

4. catkin\_make

5. git log --oneline –decorate //아래와 같이 나오면 성공

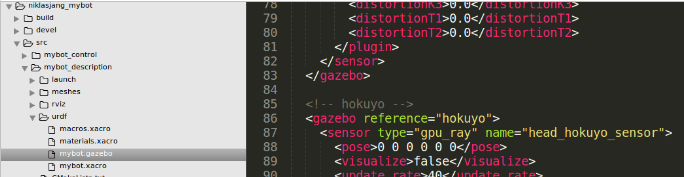


6. git checkout navigation

7. pkg\_name/src/urdf/mybot.gazebo에서 수정

#87 gpu\_ray → ray

#89 <visualize>true</visualize>

#115 libgazebo\_ros\_gpu\_laser → libgazebo\_ros\_laser

2) Layer로 맵 인식해서 rviz에서 시각화하기

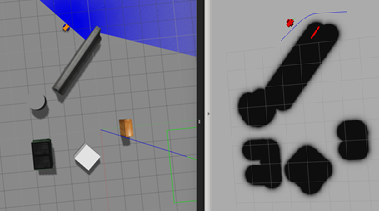
C:\Users\maker\AppData\Roaming\PolarisOffice\ETemp\11992_19602888\poclip1\26\image8.png

그림 5 모델이 촬영한 시각데이터로 만든 맵 데이터

1. roslaunch mybot\_gazebo mybot\_world.launch

2. roslaunch mybot\_navigation gmapping\_demo.launch

3. roslaunch mybot\_description mybot\_rviz\_gmapping.launch

4. roslaunch mybot\_navigation mybot\_teleop.launch

3) 인식한 맵 저장하기

1. mybot\_ws/src/mybot\_navigation에 maps/ 폴더 만들기

2. rosrun map\_server map\_saver -f ~/mybot\_ws/src/mybot\_navigation/maps/test\_map

3. ls

4. test\_map.pgm test\_map.yaml 확인

4) 저장된 맵 로드하기

1. 모든 terminal 종료

2. roslaunch mybot\_gazebo mybot\_world.launch

3. roslaunch mybot\_navigation amcl\_demo.launch

4. roslaunch mybot\_description mybot\_rviz\_amcl.launch

<https://github.com/niklasjang/path_planning>

0) Build 과정 이해하기

1. git clone

2. CmakeList주석 읽기

1) path planning pkg 만들기

2. roscore

3. rosrun niklasjang\_path\_planning path\_publisher

4. rosrun niklasjang\_path\_planning path\_subscriber2. rosrun map\_server map\_saver -f ~/catkin\_ws/src/niklasjang\_mybot/src/mybot\_navigation/maps/test\_map

3. ls

4. test\_map.pgm test\_map.yaml 확인

4) 저장된 맵 로드하기

1. 모든 terminal 종료

2. roslaunch mybot\_gazebo mybot\_world.launch

3. roslaunch mybot\_navigation amcl\_demo.launch

4. roslaunch mybot\_description mybot\_rviz\_amcl.launch

<https://github.com/niklasjang/path_planning>

0) Build 과정 이해하기

1. git clone

2. CmakeList주석 읽기

1) path planning pkg 만들기

2. roscore

3. rosrun niklasjang\_path\_planning path\_publisher

4. rosrun niklasjang\_path\_planning path\_subscriber

**2-2 PD control의 대안으로서의 diff-drive-plugin**

1. PD control은 현재의 위치를 계속해서 받아와야하므로, 실제 룸바를 가지고 path-planning을 진행할 때 현재 위치를 정확하지 받아오지 못하면 최종 결과에서 오차가 누적될 것으로 예상된다. 따라서 기본 diff\_drive\_pulgin을 사용해서 기본적인 주행을 하고, 부가적으로 PD Control을 사용하는 방향으로 진행하고 있다.

2.diff\_contorl\_plugin은 linear x,y,z와 angular x,y,z를 입력으로 받는 geometry/Twist msg를 전달받는다. 6개의 component중에서 2차원 좌표의 위치를 의미하는 linear의 x,y와 yaw를 담당하는 angular의 z만을 사용한다. 특히, z의 값이 작을 경우에는 양 바퀴가 같은 방향으로 회전하며, 적절히 큰 값을 가질 때에는 제자리에서 회전을 하기 위해 양 바퀴가 서로 다른 방향으로 회전한다.

3. 현재 diff\_control\_plgun을 사용해서 PDDL의 결과를 수행하기 위해 robot에게 ‘회전-직진’으로 구성된 적절한 명령을 보내는 custom plugin을 작성 중이다.

**3. Issue 및 대책**

Q1. 현재 많은 사람들이 diff\_drive\_control\_plugin으로 90degree 회전을 하기 위해서 일정 시간동안 회전하도록 명령하는 방법을 사용한다. 현실에서 이 방법에서 어느정도의 오차가 발생할지 알 수 없다.

A1. 기본적인 diff\_drive\_control\_plugin에 더해 PD control을 사용해서 오차를 줄이는 방향으로 개발할 계획이다.