산학프로젝트 주간 보고서 ( 2019. 03. 28)

이름/팀명 : 장 환석/AP 프로젝트명 : AI planning

**1.개인별 주간 목표(상세 개발 계획서)**

|  |  |
| --- | --- |
| 1주차 | 프로젝트 설명회 |
| 2주차 | 팀 빌딩 확정, 프로젝트 제안서 논의 |
| 3주차 | 0. 물품 구매 요청서 및 사유서 작성  1. ROS 환경 이해 및 ROS 기본 topic publish/subscribe 구현  2. Rviz, Gazebo 기본 개념 이해 |
| 4주차 | 1. Simulator 자동차 모델링  2. ROS 노드와 Gazebo 노드의 pub/sub 구현  3. ROS parameter server 이해 및 사용 |
| 5주차 | 1. Rviz navigation이해 및 분석  2. Rviz manipulation 이해 및 분석  3. Rviz와 Gazebo 연동 이해 |
| 6주차 | 1. A\* 알고리즘과 PDDL 이해 및 개발  2. 시뮬레이터 연동 |
| 7주차 | 1. A\* 알고리즘과 PDDL 이해 및 개발  2. 시뮬레이터 연동 |
| 8주차 | 1. AI path-planning 알고리즘 연구 |
| 9주차 | 1. AI path-planning 알고리즘 연구 |
| 10주차 | 1. AI path-planning 알고리즘 연구 |
| 11주차 | 1. AI path-planning 시뮬레이터와 연동 |
| 12주차 | 1. AI path-planning 시뮬레이터와 연동 |
| 13주차 | 1. AI path -planning irobot roomba와 연동 |
| 14주차 | 1. AI path -planning irobot roomba와 연동 |
| 15주차 | 1. AI path -planning irobot roomba와 Web 연동 |

**2. 추진 실적**

0)Irobot create 2 및 raspberry pi 구매요청서 및 사유서 작성

본 프로젝트에 필요한 물품에 대한 구매요청서와 사유서를 학과 사무실을 통해서 학과장님께 전달하였습니다.

1) ROS 환경 이해 및 기본 topic pub/sub 구현

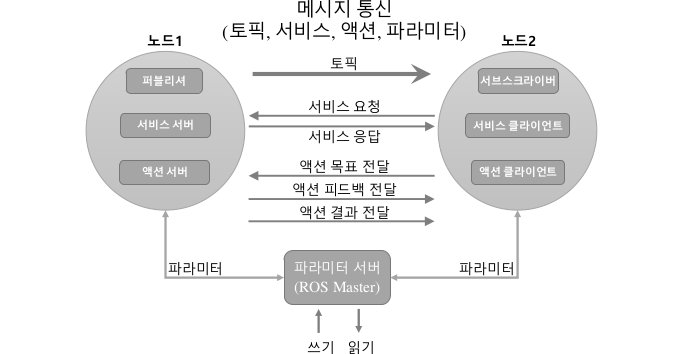
-ROS 환경 이해

1. 토픽 : 노드1에서 노드2로의 메시지 통신

2. 서비스 : 두 노드 간의 서비스 요청 및 서비스 응답 통신

3. 액션 : 액션 목표 전달, 액션 피드백 전달(중간 결과), 액션 결과 전달

4. 파라미터 서버 : ROS master를 통한 프로그램 종료 없이 param값을 변경해서 진행 가능



-기본 topic pub/sub 구현

1. \*catkin 작업공간 만들기 : \*빌드를 위한 환경 설정들이 저장되어있는 CmakeList.txt를 생성

\*빌드 : raw source data를 end user가 사용할 target 포멧으로 만드는 것

\*catkin :rosbuild의 후임자로서 범용 빌드 환경인 CMAKE보다 더욱 ROS에 최적화

2. catkin 빌드하기 : build와 devel 디렉토리가 생성된다. Devel/setup.bash(환경설정파일)를 ROS 환경의 최상위에 오버레이 해서 이 작업 공간을 사용할 것을 명시한다.

3. ROS File System

package를 만들면 크게 아래의 두 파일이 생성됩니다.

Packages: 패키지는 ROS 코드의 소프트웨어 구조 단위입니다. 각각의 패키지는 라이브러리, 실행파일, 스크립트 등을 담고 있습니다.

Manifest ([package.xml](http://wiki.ros.org/catkin/package.xml)): 매니패스트는 패키지에 대한 설명을 기록한 것입니다. 이것은 패키지간의 의존성과 패키지의 버전, 관리자, 라이센스 등의 메타 정보를 관리하는 데 쓰입니다.

4. ROS package 만들기 및 package 구조 이해하기

package는 아래의 두 파일을 반드시 가져야합니다. 그리고 한 파일당 하나의 package만이 허용됩니다. 내부 package 같은 경우는 존재하지 않습니다.

my\_package/

CMakeLists.txt

package.xml

각각의 package들은 아래와 같이 관리됩니다.

workspace\_folder/ -- 작업공간

src/ -- 소스 폴더

CMakeLists.txt -- catkin이 제공하는 '최상위'의 CMake 파일,

package\_1/

CMakeLists.txt -- package\_1에 대한 CMakeLists.txt 파일

package.xml -- package\_1에 대한 매니패스트

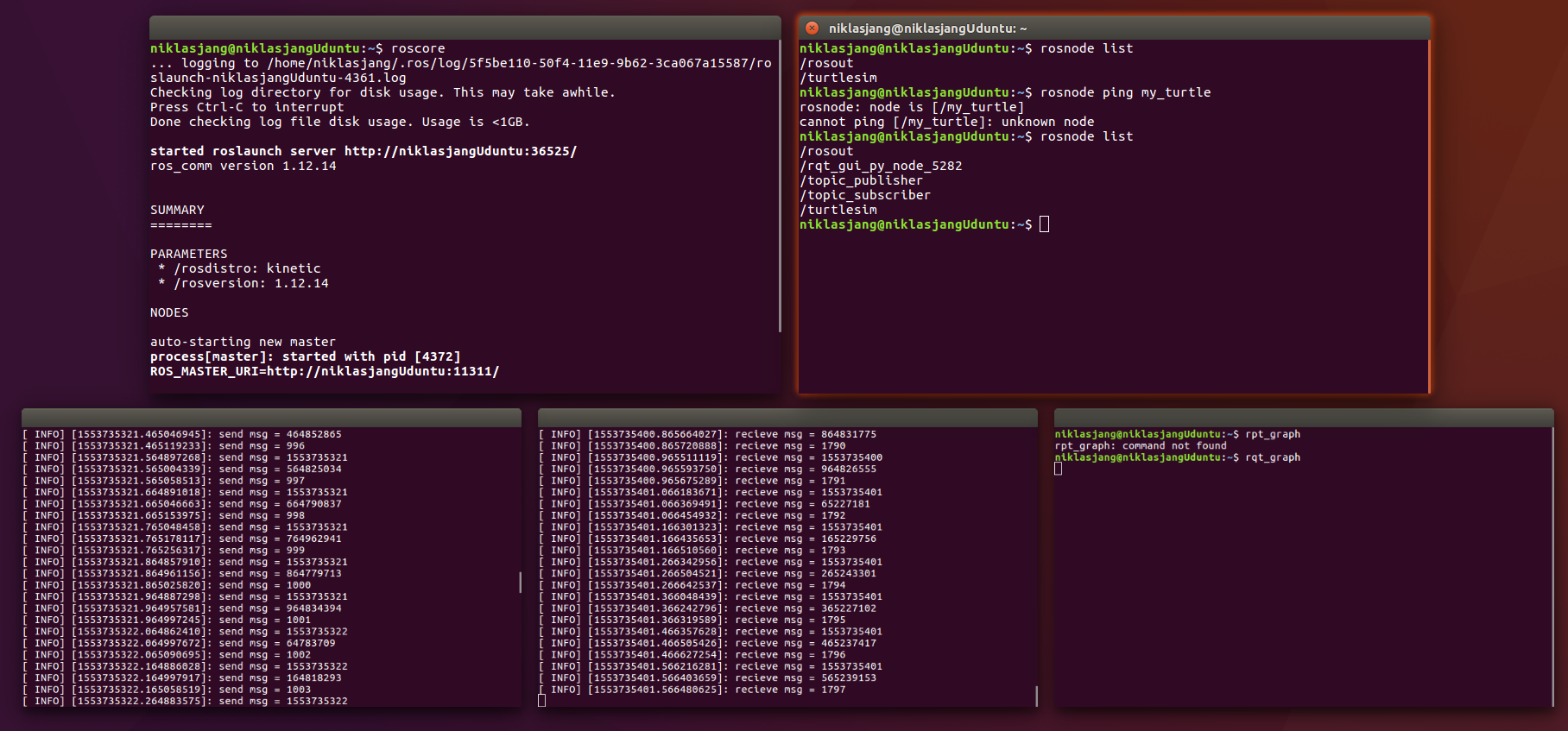
...

package\_n/

CMakeLists.txt -- package\_n에 대한 CMakeLists.txt 파일

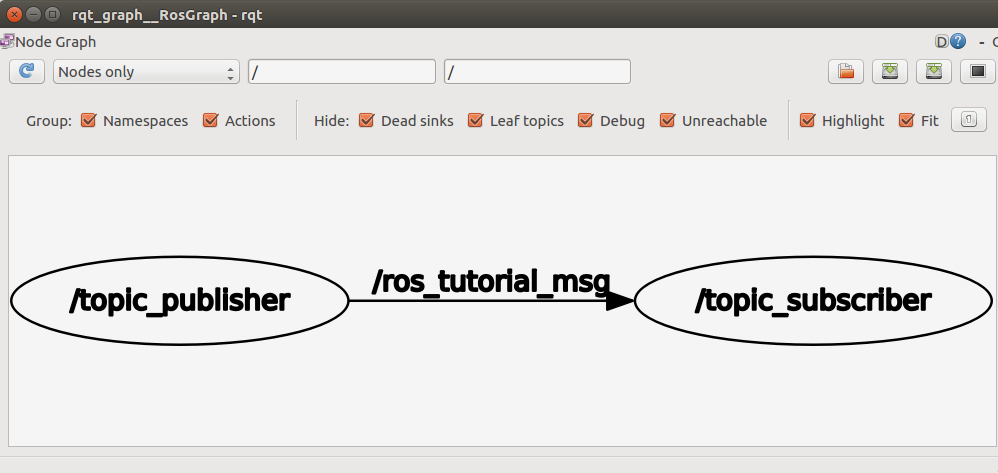
package.xml -- package\_n에 대한 매니패스트

5. Ros 기본 topic publish/subscribe



상단 왼쪽 : roscore, 상단 오른쪽: rosnode list

하단 왼쪽 : publisher 하단 중간 : subscriber 하단 오른쪽: rqt\_graph



3) Rviz와 Gazebo 기본 개념 이해

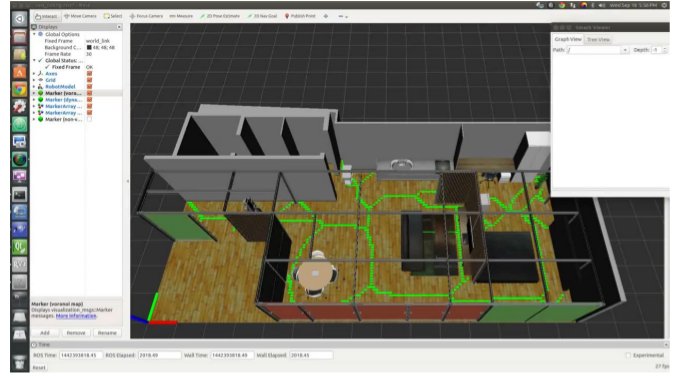
-Rviz : ROS의 3D 시각화

1. 강력한 3D 시각화툴 제공

2. 레이저, **카메라 등의 센서 데이터를 시각화**

**3. 로봇 외형과 계툴획된 동작을 표현 (** URDF（Unified Robot Description Format）)

4. 매니퓰레이션과 네비게이션 : Rviz환경에 문제 해결 환경을그대로 mapping한 뒤, robot model을 추가하여 두 지점 간의 path planning을 시각화해서 볼 수 있다.

5. 원격제어****

-RQT : GUI툴을 제공

1. 그래픽 인터페이스 개발을 위한 Qt 기반 프레임 워크 제공 : 본 업무에 집중할 수 있게 도움

**2. 노드와 그들 사이의 연결 정보 표시(rqt\_graph)**

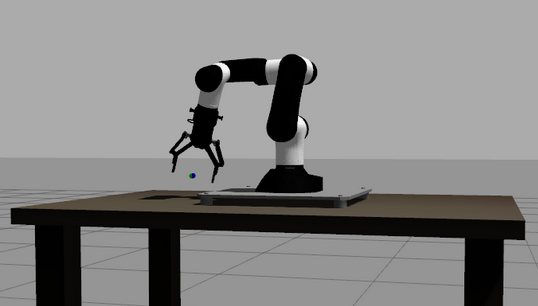
3. 인코더, 전압, 또는 시간이 지남에 따라 변화하는 숫자를 플로팅(rqt\_plot)

4. 데이터를 메시지 형태로 기록하고 재생(rqt\_bag)

-Gazebo

1. 물리 엔진을 탑재, 로봇, 센서, 환경 모델 등을 지원하는 3차원 시뮬레이터

2. ROS와의 높은 호완성



3. Issue 및 대책

Q1. 팀원 간 코드 공동 개발 환경

A1. Git bash 와 Git hub를 사용한 분산 버전 관리 시스템 적용

Q2. Rviz와 Gazebo 중 본 프로젝트에 적합한 프로그램이 무엇인지 불분명

A2. 4주차와 5주차의 상세 개발 계획에 따라서 이들 간의 차이점을 명확히 인지 후 결정

Q3. 필요 물품 구매요청서의 반려 가능성

A3. 시뮬레이터 환경을 확실히 이해하여 차선책으로서 구현 계획을 세움

Q4. 개발 상세 계획에 대한 불확실성과 불명확성

A4. 연구 과정을 통해서 차기 상세 계획들을 보다 자세하게 재구성할 계획

Q5. 개발 언어

A5. C++와 python을 모두 염두해두고 있으며, 초기 결과물의 완성 시점과 최종 결과물의 최적화를 고려해서 가치 선택할 계획