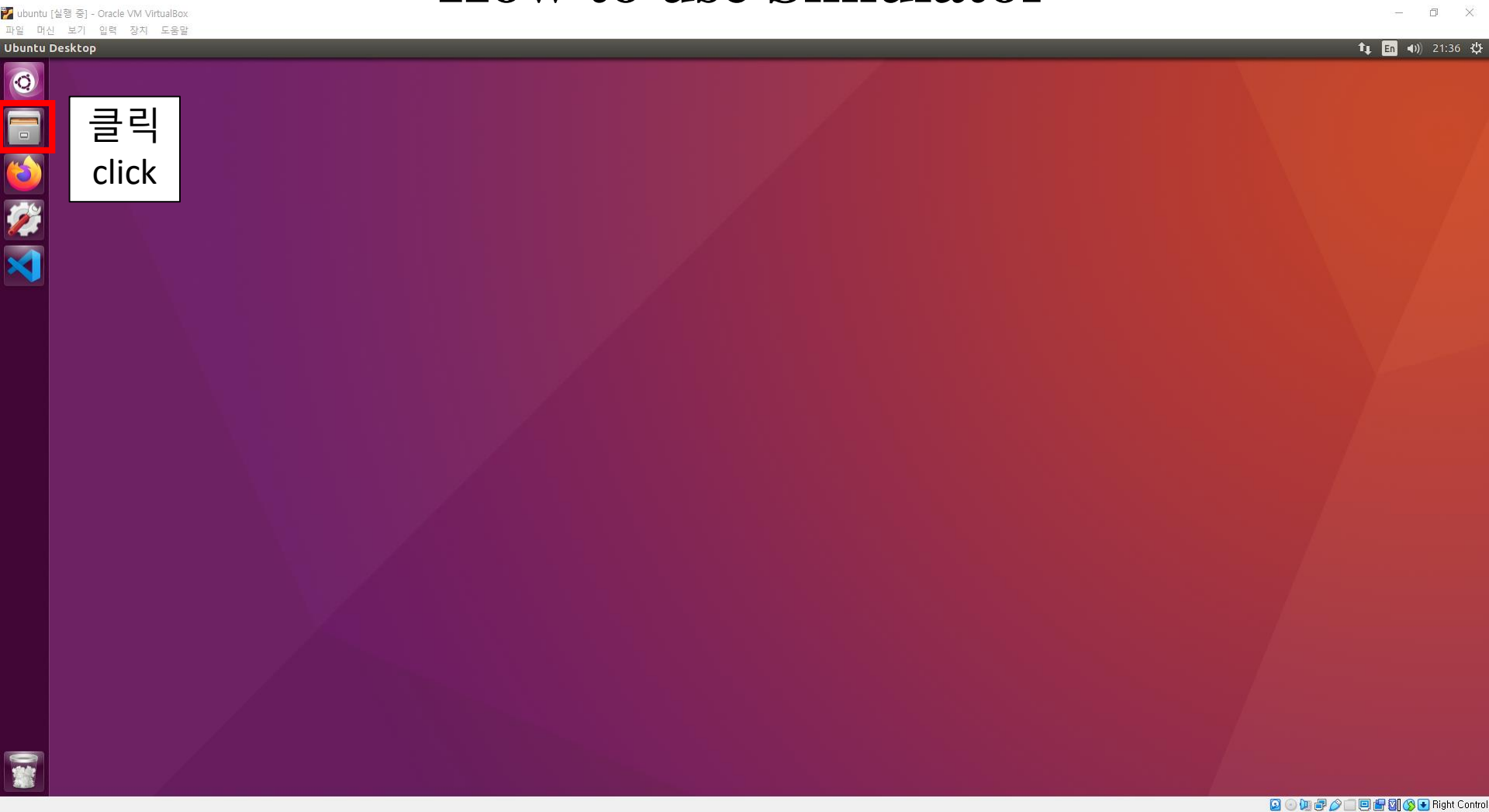


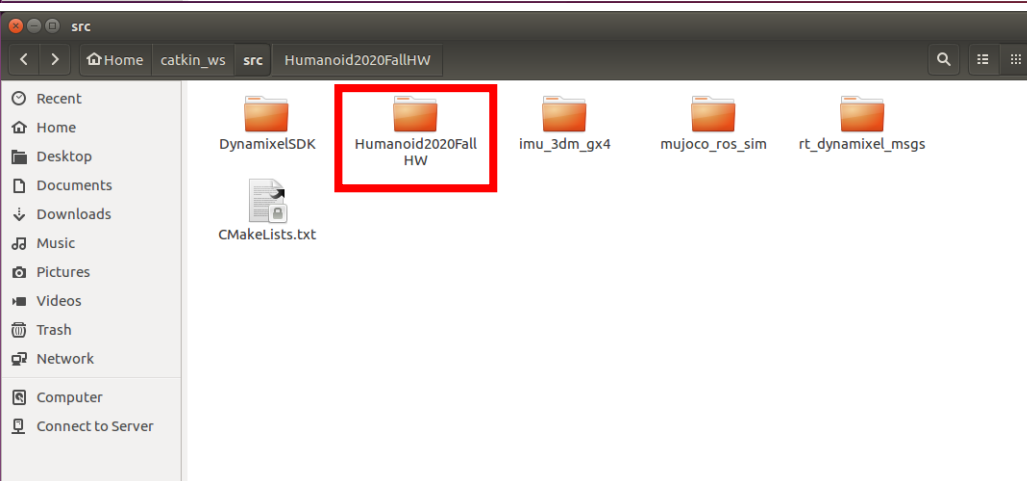
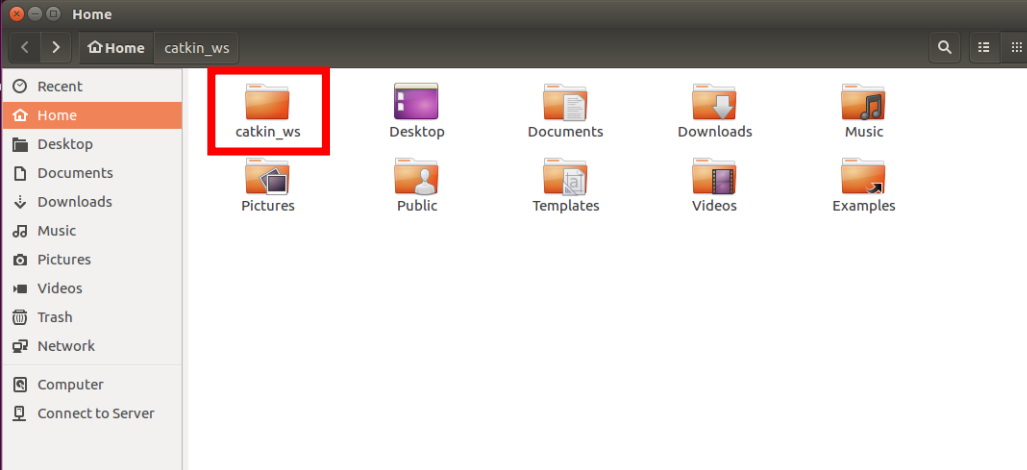
# **시뮬레이터 설명**

## **How to use Simulator**

# 시뮬레이터 설명

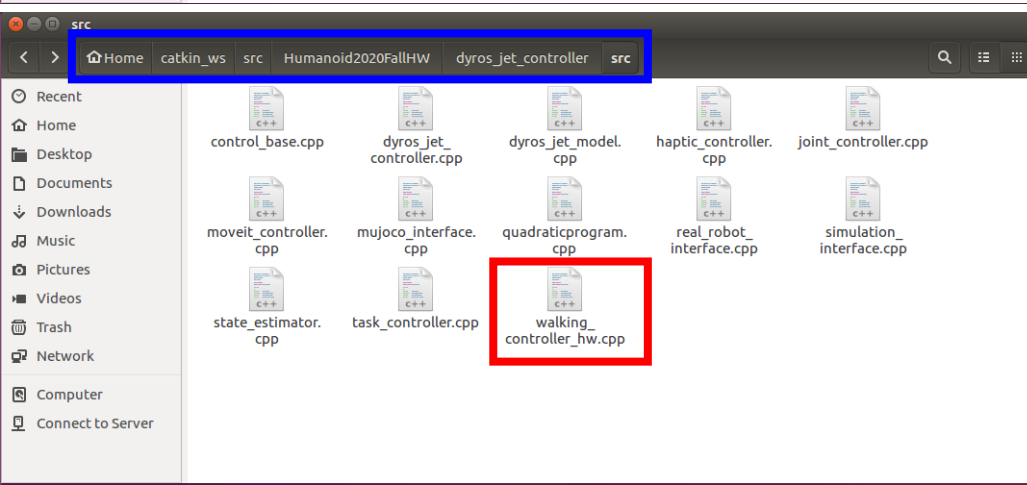
## How to use Simulator

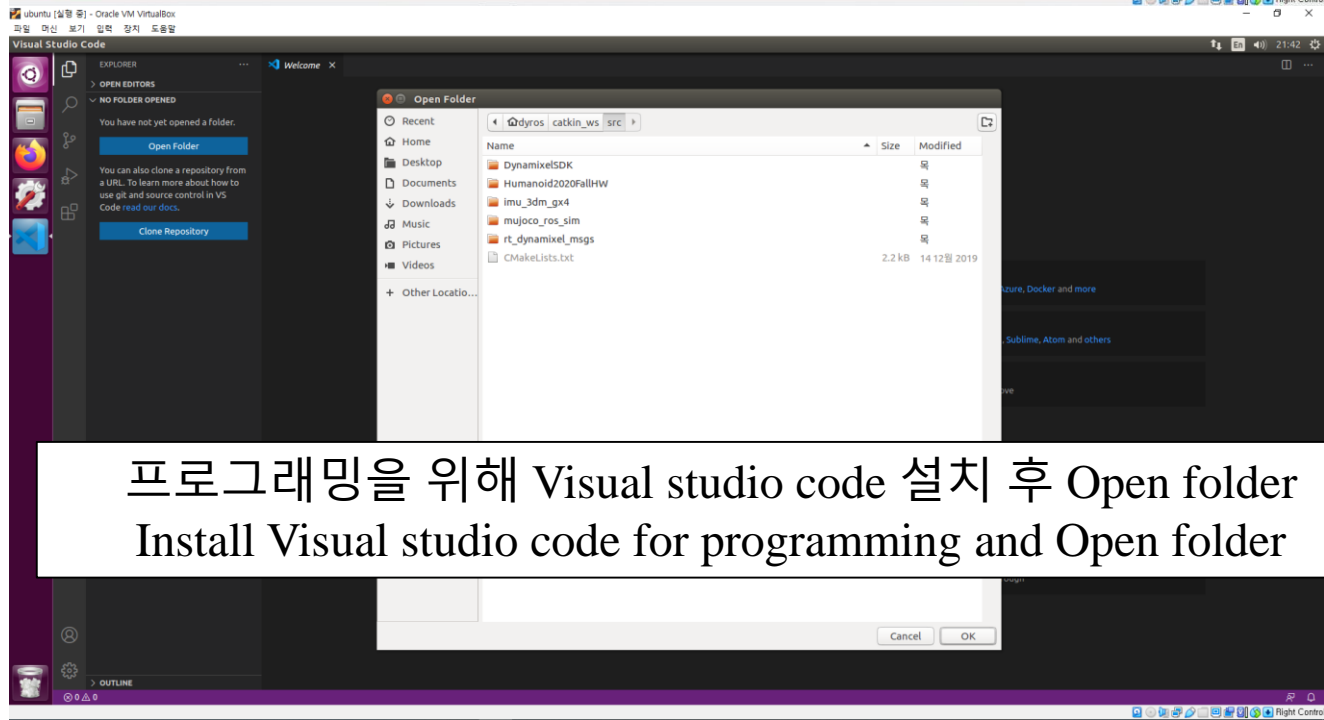
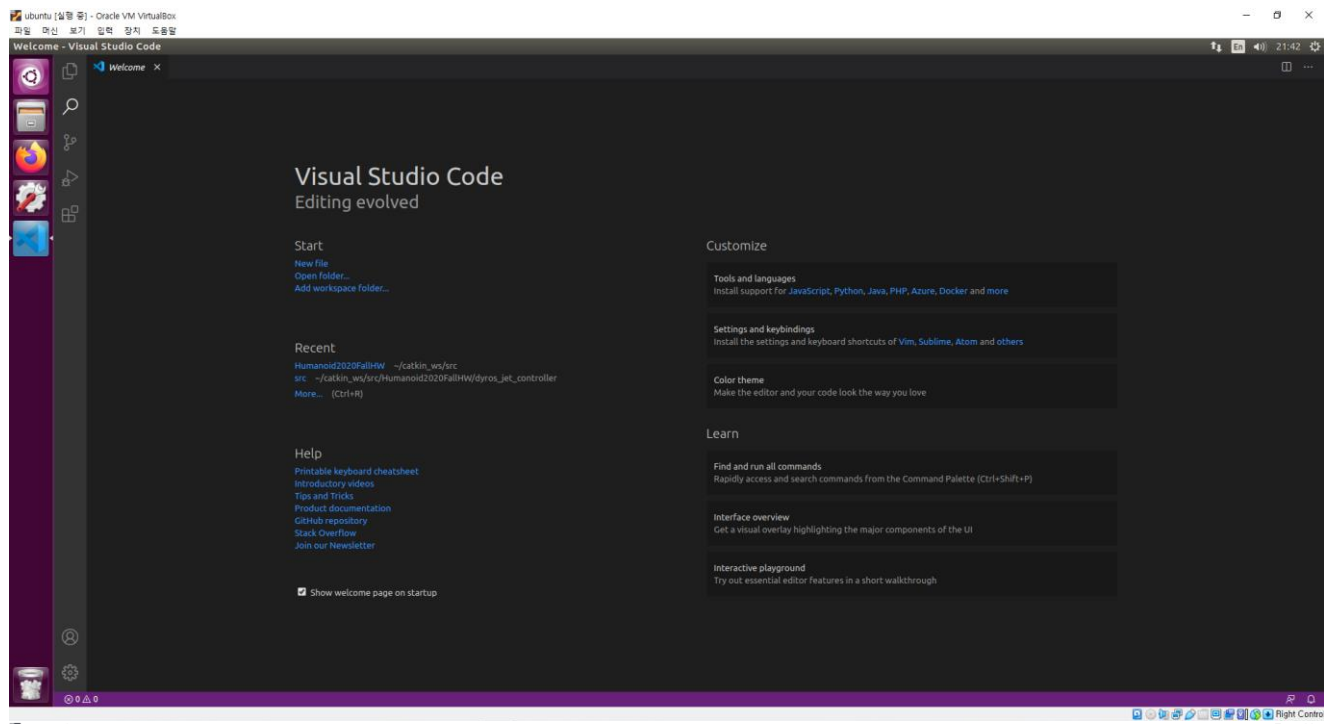




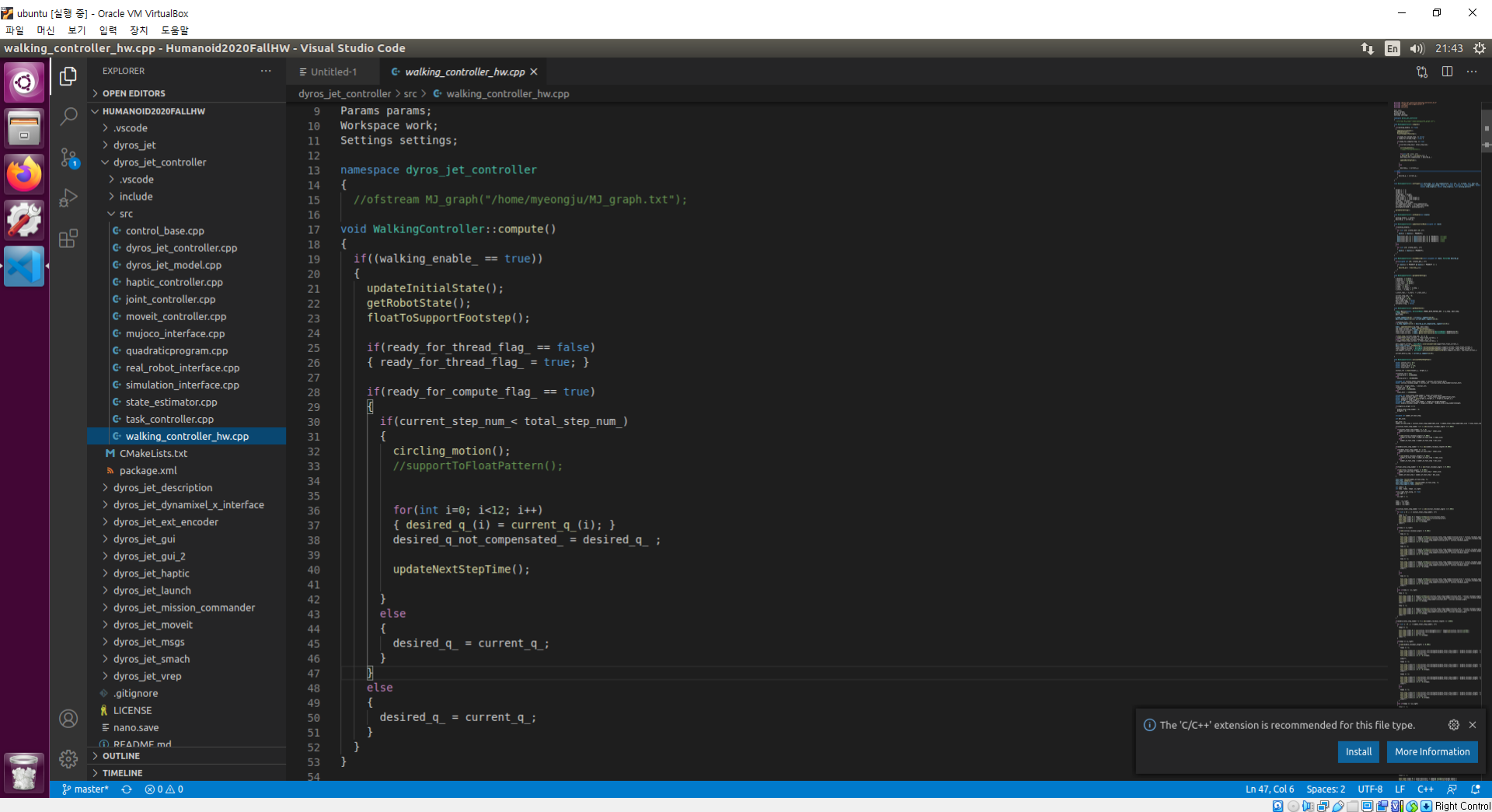
해당 위치에 과제에 활용될 cpp  
파일 확인

Check cpp file for HW





프로그래밍을 위해 Visual studio code 설치 후 Open folder  
Install Visual studio code for programming and Open folder



한학기 동안 사용할 코드 프레임워크  
Code Framework for Semester

# 빌드하는 방법

## How to build

ubuntu [실행 중] - Oracle VM VirtualBox  
파일 마신 보기 입력 장치 도움말

The screenshot shows a VS Code editor with a project named 'HUMANOID2020FALLHW'. The file explorer on the left shows the project structure, including 'src' and 'include' directories. The main editor window displays the code for 'walking\_controller\_hw.cpp'. The code defines a 'WalkingController' class with methods for state management and motion planning. A terminal window is open in the foreground, showing the command 'catkin\_make' being executed in the 'catkin\_ws' directory. The terminal output shows the build process for the 'walking\_controller' package.

```
namespace dyros_jet_controller
{
    //ofstream MJ_graph("/home/myeongju/MJ_graph.txt");

    void WalkingController::compute()
    {
        if((walking_enable_ == true))
        {
            updateInitialState();
            getRobotState();
            floatToSupportFootstep();

            if(ready_for_thread_flag == false)
            { ready_for_thread_flag = true; }

            if(ready_for_compute_flag == true)
            {
                if(current_step_num < total_step_num_)
                {
                    circling_motion();
                    //supportToFloatPattern();

                    for(int i=0; i<12; i++)
                    { desired_q(i) = current_q(i); }
                    desired_q_not_compensated_ = desired_q ;

                    updateNextStepTime();
                }
                else
                {
                    desired_q = current_q ;
                }
            }
            else
            {
                desired_q = current_q ;
            }
        }
    }

    void WalkingController::setTarget(int walk mode, bool hip compensation, bool lqr, int ik mode, bool heel_toe,
    bool is_right_foot_swing, double x, double y, double z, double height, double theta,
    double step_length_x, double step_length_y, bool walking pattern)
    {
        // TODO: Implement setTarget function
    }
}
```

Terminal output:

```
dyros@dyros: ~/catkin_ws
dyros@dyros:~$ cd catkin_ws/
dyros@dyros:~/catkin_ws$ catkin_make
```

코드를 수정한 후 Ctrl + S를 눌러서 저장한 다음 빌드  
Save by Ctrl + S after editing. Build by typing "catkin\_make"

EXPLORER

OPEN EDITORS

dyros\_jet\_controller > src > walking\_controller\_hw.cpp

HUMANOID2020FALLHW

.vscode

dyros\_jet

dyros\_jet\_controller

.vscode

include

src

control\_base.cpp

dyros\_jet\_controller.cpp

dyros\_jet\_model.cpp

haptic\_controller.cpp

joint\_controller.cpp

moveit\_controller.cpp

mujoco\_interface.cpp

quadraticprogram.cpp

real\_robot\_interface.cpp

simulation\_interface.cpp

state\_estimator.cpp

task\_controller.cpp

walking\_controller\_hw.cpp

CMakeLists.txt

package.xml

dyros\_jet\_description

dyros\_jet\_dynamixel\_x\_interface

dyros\_jet\_ext\_encoder

dyros\_jet\_gui

dyros\_jet\_gui\_2

dyros\_jet\_haptic

dyros\_jet\_launch

dyros\_jet\_mission\_commander

dyros\_jet\_moveit

dyros\_jet\_msgs

dyros\_jet\_smach

dyros\_jet\_vrep

.gitignore

LICENSE

nano.save

README.md

OUTLINE

TIMELINE

dyros\_jet\_controller > src > walking\_controller\_hw.cpp

11 Settings Settings;

12

13 namespace dyros\_jet\_controller

14 {

15 //ofstream MJ\_graph("/home/myeongju/MJ\_graph.txt");

16

17 void WalkingController::compute()

18 {

19 if((walking\_enable\_ == true))

20 {

21 updateInitialState();

22 getRobotState();

23 floatToSupportFootstep();

24

25 if(ready\_for\_thread\_flag\_ == false)

26 { ready\_for\_thread\_flag\_ = true; }

27

28 if(ready\_for\_compute\_flag\_ == true)

29 {

30 if(current\_step\_num < total\_step\_num\_)

31 {

32 circling\_motion();

33 //supportToFloatPattern();

34

35 for(int i=0; i<12; i++)

36 { desired\_q(i) = current\_q(i); }

37 desired\_q\_not\_compensated\_ = desired\_q ;

38

39 updateNextStepTime();

40

41 }

42 }

43 else

44 {

45 desired\_q\_ = current\_q ;

46 }

47 }

48 else

49 {

50 desired\_q\_ = current\_q ;

51 }

52 }

53 }

54 }

55 void WalkingController::setTarget(int walk mode, bool hip compensation, bool lqr, int ik mode, bool heel toe,

56 bool is\_right foot swing, double x, double y, double z, double height, double theta,

57 double step length x, double step length y, bool walking pattern)

dyros@dyros: ~/catkin\_ws

[ 85%] Built target imu\_3dm\_gx4\_generate\_messages\_lisp

[ 85%] Built target tf2\_msgs\_generate\_messages\_nodejs

[ 85%] Built target tf2\_msgs\_generate\_messages\_eus

[ 85%] Built target tf2\_msgs\_generate\_messages\_py

[ 86%] Built target imu\_3dm\_gx4\_generate\_messages\_py

[ 86%] Built target tf2\_msgs\_generate\_messages\_lisp

[ 88%] Built target imu\_3dm\_gx4

[ 88%] Built target imu\_3dm\_gx4\_generate\_messages

Scanning dependencies of target dyros\_jet\_controller\_lib

[ 89%] Building CXX object Humanoid2020FallHW/dyros\_jet\_controller/CMakeFiles/dyros\_jet\_controller\_lib.dir/src/walking\_controller\_hw.cpp.o

[ 90%] Linking CXX shared library /home/dyros/catkin\_ws/devel/lib/libdyros\_jet\_controller\_lib.so

[ 96%] Built target dyros\_jet\_controller\_lib

[ 97%] Linking CXX executable /home/dyros/catkin\_ws/devel/lib/dyros\_jet\_controller/dyros\_jet\_controller

[ 97%] Built target dyros\_jet\_controller

[ 97%] Built target tf\_generate\_messages\_cpp

[ 97%] Built target tf\_generate\_messages\_nodejs

[ 97%] Built target tf\_generate\_messages\_lisp

[ 97%] Built target tf\_generate\_messages\_py

[ 97%] Built target tf\_generate\_messages\_eus

[100%] Built target x\_interface

dyros@dyros:~/catkin\_ws\$

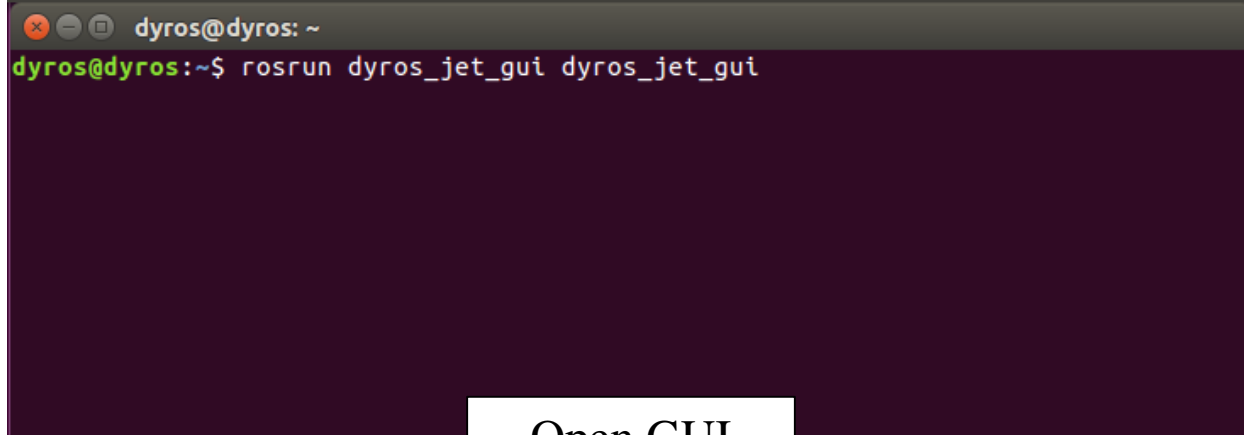
빌드 완료  
Build complete

```
dyros@dyros: ~  
dyros@dyros:~$ roscore
```

## Roscore

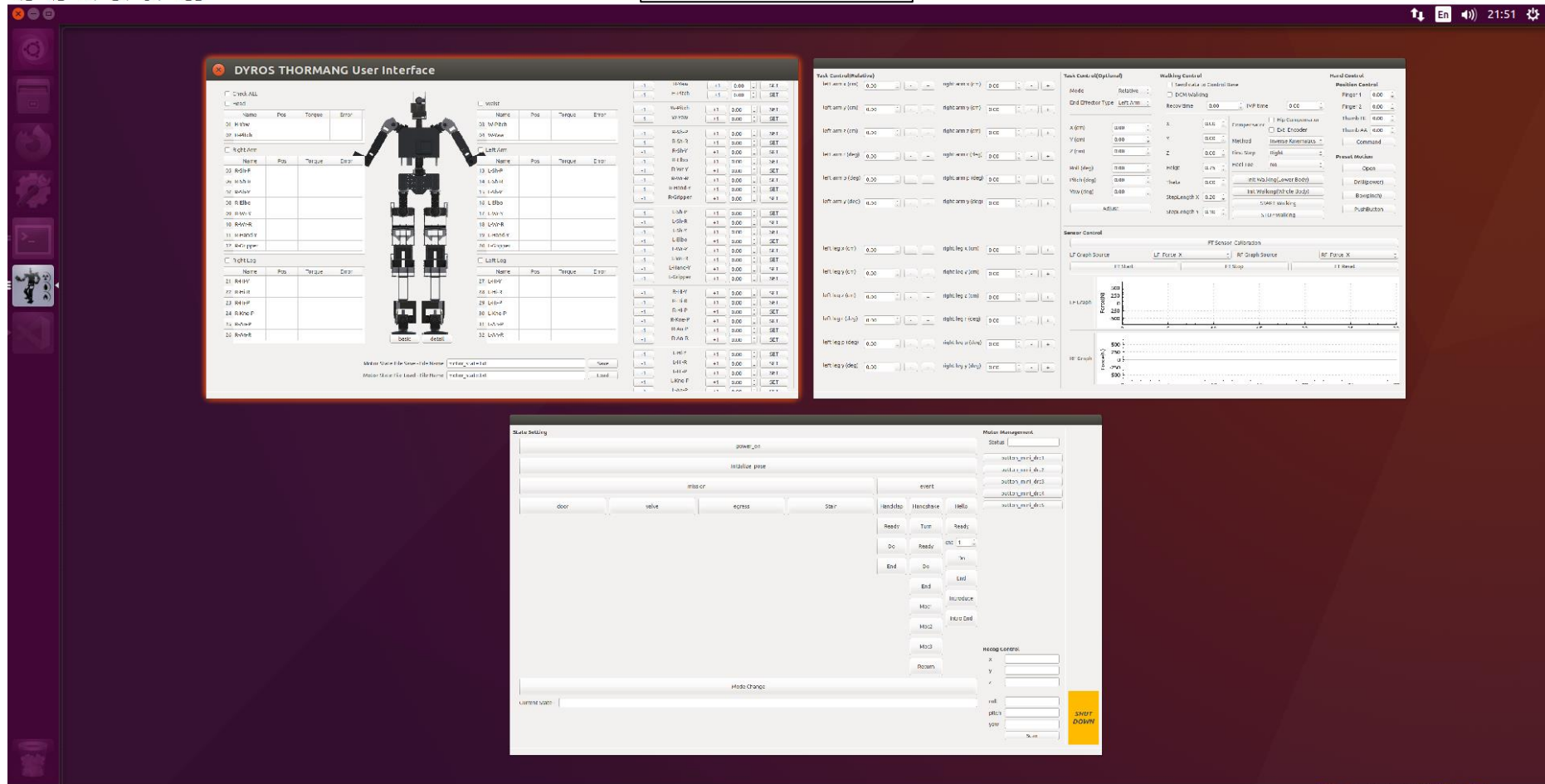
```
roscore http://dyros:11311/  
Press Ctrl-C to interrupt  
Done checking log file disk usage. Usage is <1GB.  
  
started roslaunch server http://dyros:38565/  
ros_comm version 1.12.15  
  
SUMMARY  
=====  
  
PARAMETERS  
* /rostdistro: kinetic  
* /rosversion: 1.12.15  
  
NODES  
  
auto-starting new master  
process[master]: started with pid [6389]  
ROS_MASTER_URI=http://dyros:11311/  
  
setting /run_id to 353f1bd2-ec51-11ea-b6df-0800276e0851  
process[rosout-1]: started with pid [6402]  
started core service [/rosout]  
█
```

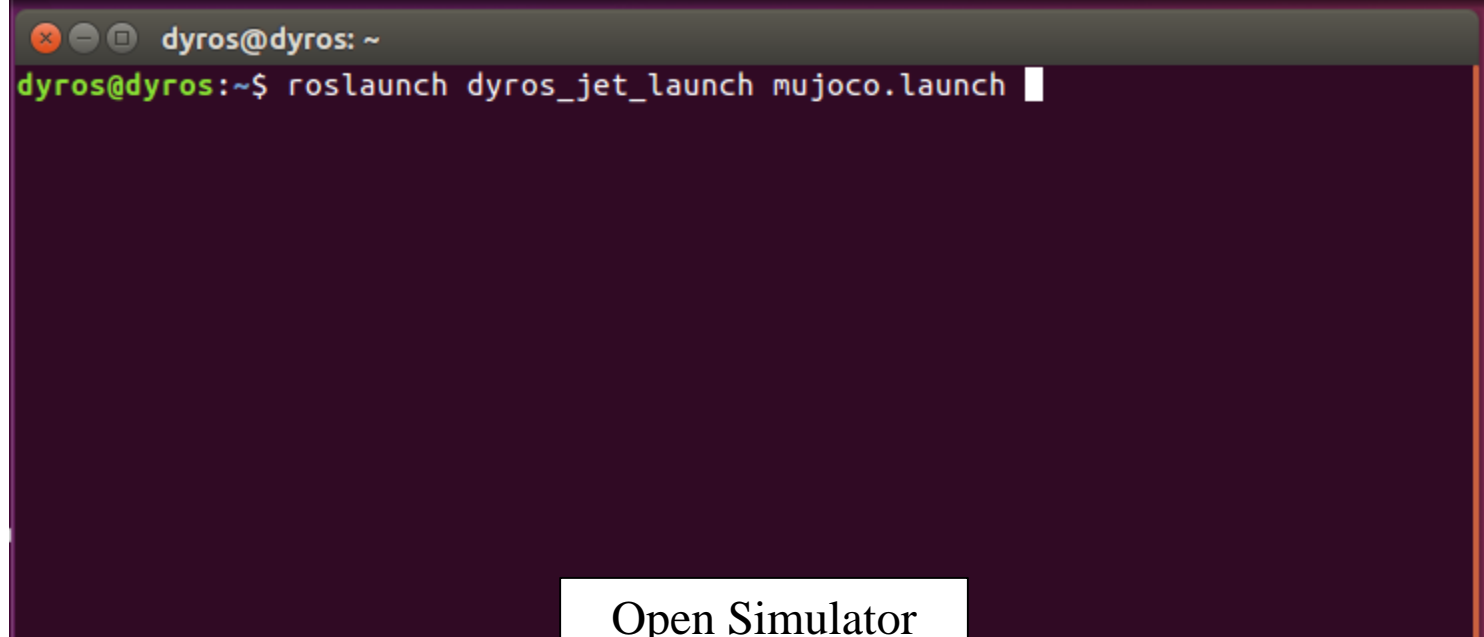




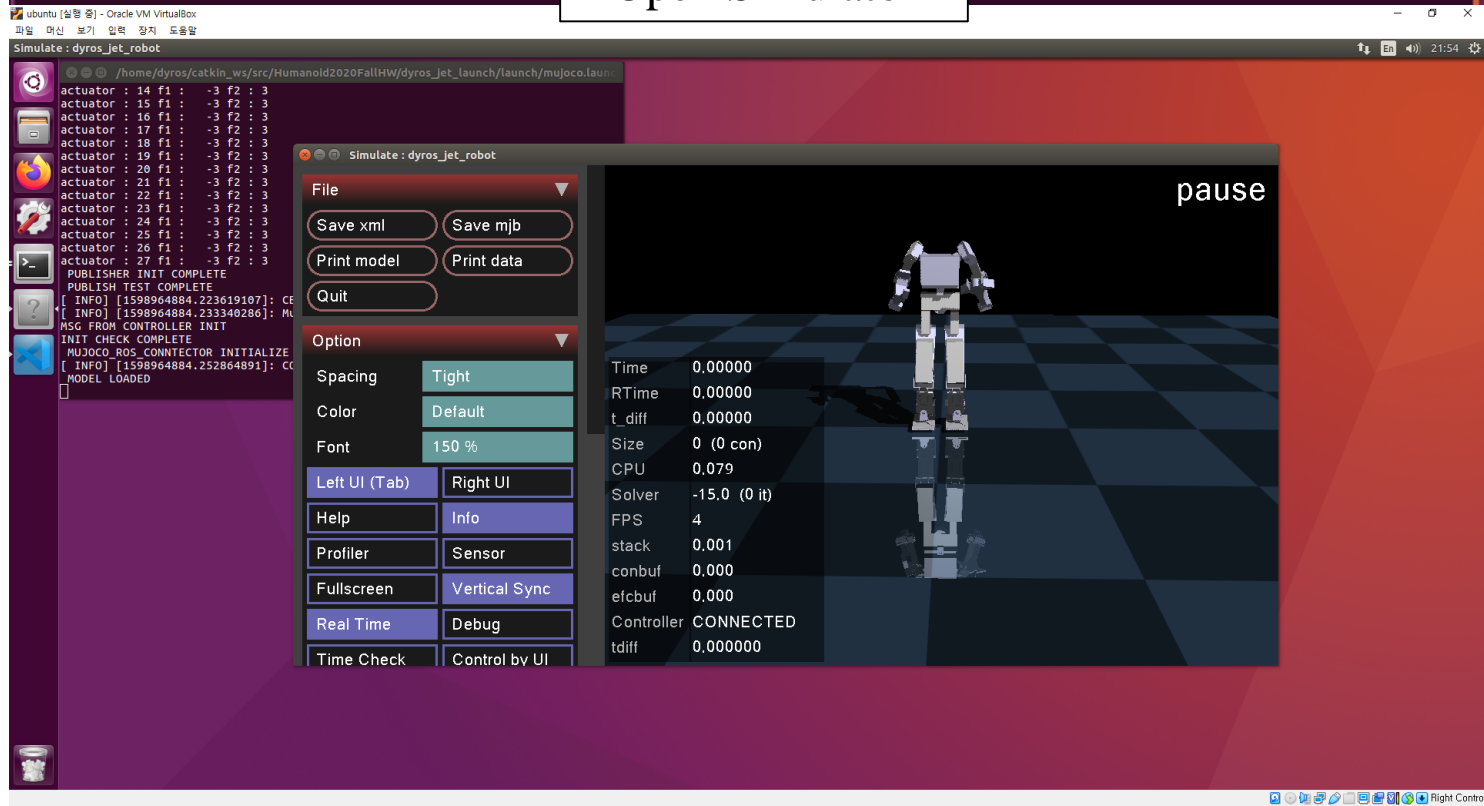
Open GUI

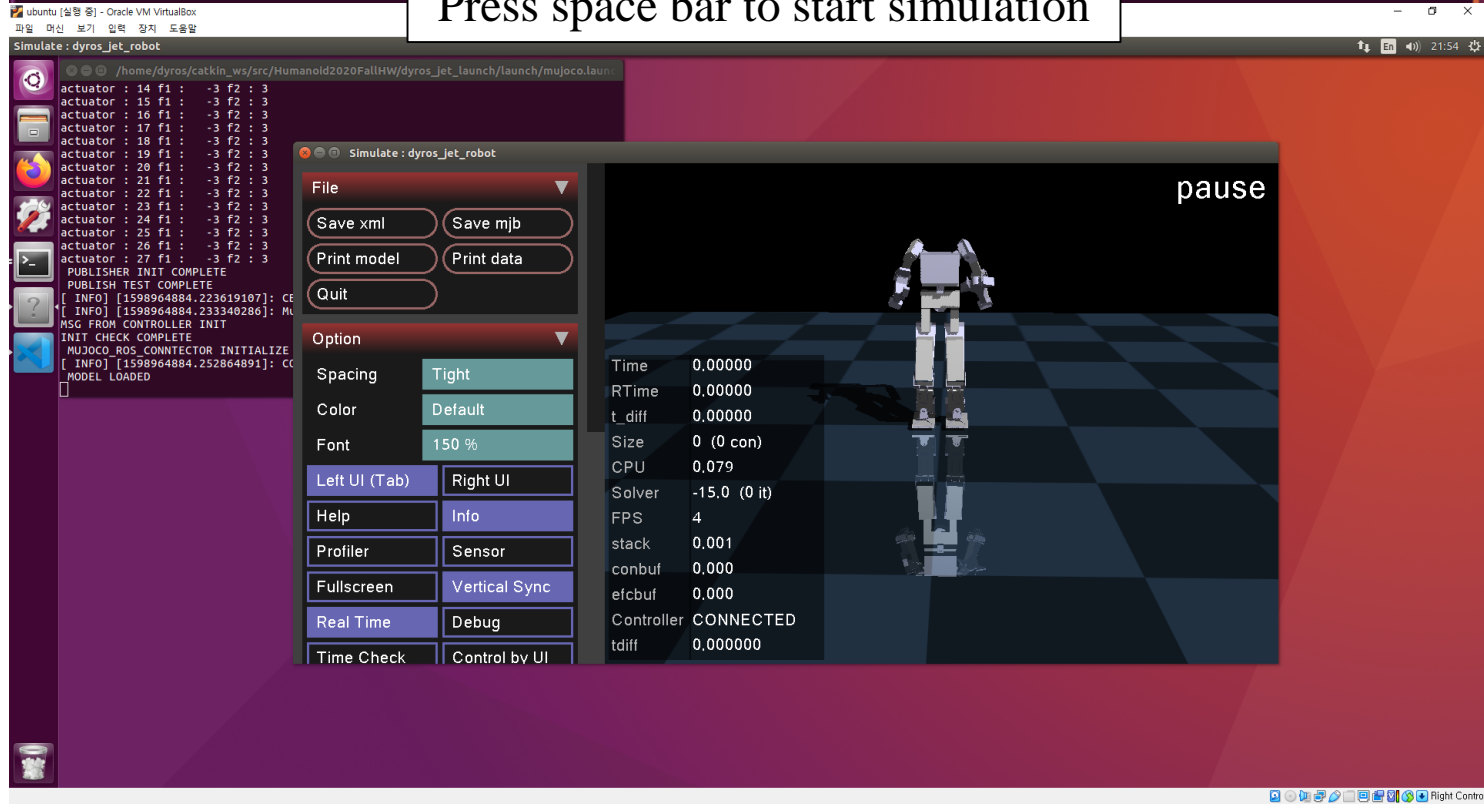
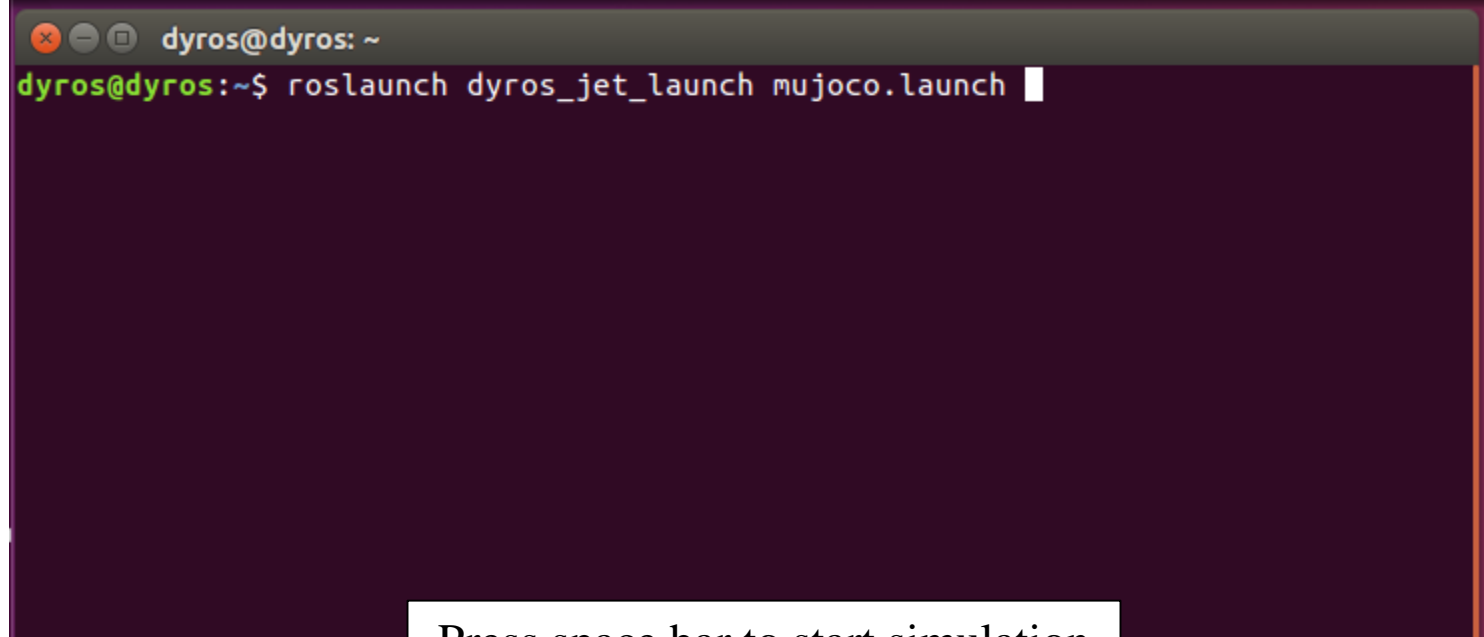
ubuntu [실행 중] - Oracle VM VirtualBox  
파일 머신 보기 입력 장치 도움말

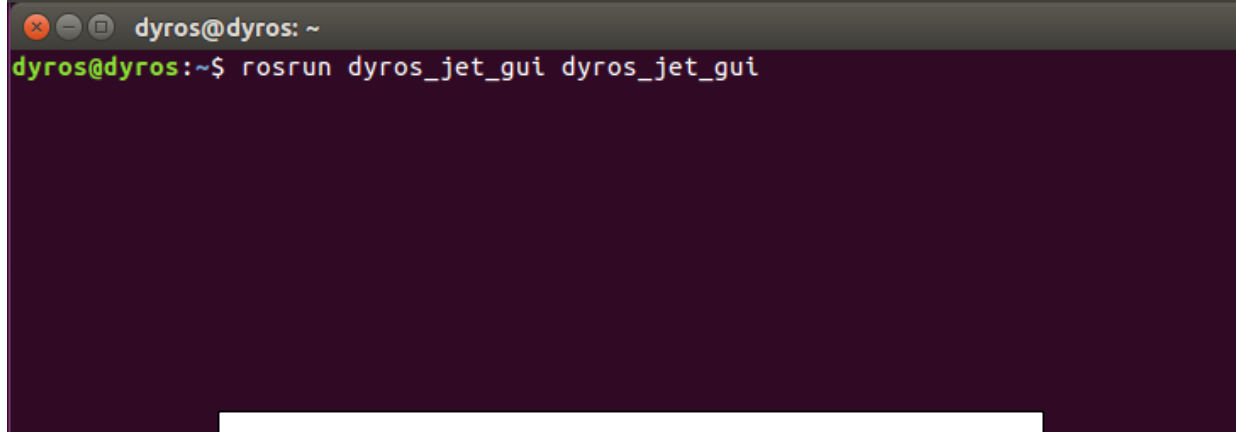




Open Simulator







Press start walking to start controller

