**环境配置**

* 安装仿真系统依赖包

sudo apt-get install -y ros-noetic-navigation ros-noetic-teb-local-planner\* ros-noetic-ros-control ros-noetic-ros-controllers ros-noetic-gazebo-ros-control ros-noetic-ackermann-msgs ros-noetic-effort-controllers ros-noetic-joint-state-controller ros-noetic-tf2-ros ros-noetic-tf

* turtlebot3\_simulation包需要搭配turtlebot3包使用
  + turtlebot3包安装：[参考链接](https://www.ncnynl.com/archives/201707/1817.html" \t "_blank)
* 安装turtlebot3\_simulation包

$ cd ~/catkin\_ws/src/

$ git clone https://github.com/ROBOTIS-GIT/turtlebot3\_simulations.git

$ cd ~/catkin\_ws && catkin\_make

* 新终端，启动仿真测试

$ export TURTLEBOT3\_MODEL=burger

## TURTLEBOT3\_MODEL有burger, waffle或waffle\_pi三种

$ roslaunch turtlebot3\_fake turtlebot3\_fake.launch

* TurtleBot3 仿真节点不依赖实体机器人，也可以在rviz里通过teleop节点进行控制
* 新终端，启动键盘控制

$ export TURTLEBOT3\_MODEL=burger

## TURTLEBOT3\_MODEL有burger, waffle或waffle\_pi三种

$ roslaunch turtlebot3\_teleop turtlebot3\_teleop\_key.launch

**加载gazebo环境**

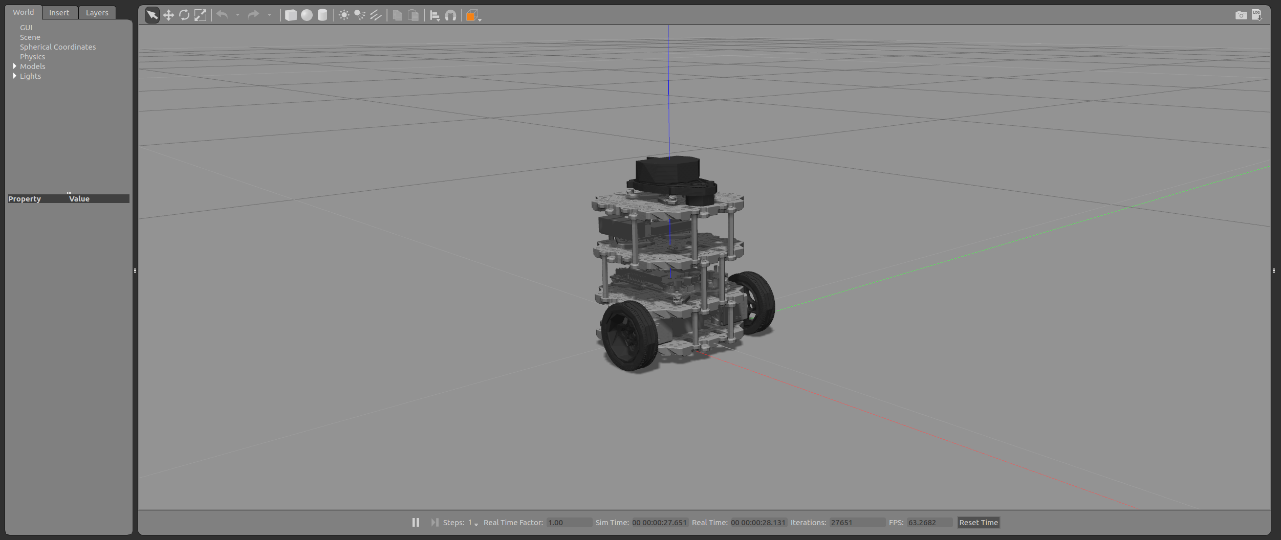
**在Gazebo加载各种环境**

* 在Gazebo加载Empty World

$ export TURTLEBOT3\_MODEL=${TB3\_MODEL}

## TURTLEBOT3\_MODEL有burger, waffle或waffle\_pi三种

$ roslaunch turtlebot3\_gazebo turtlebot3\_empty\_world.launch

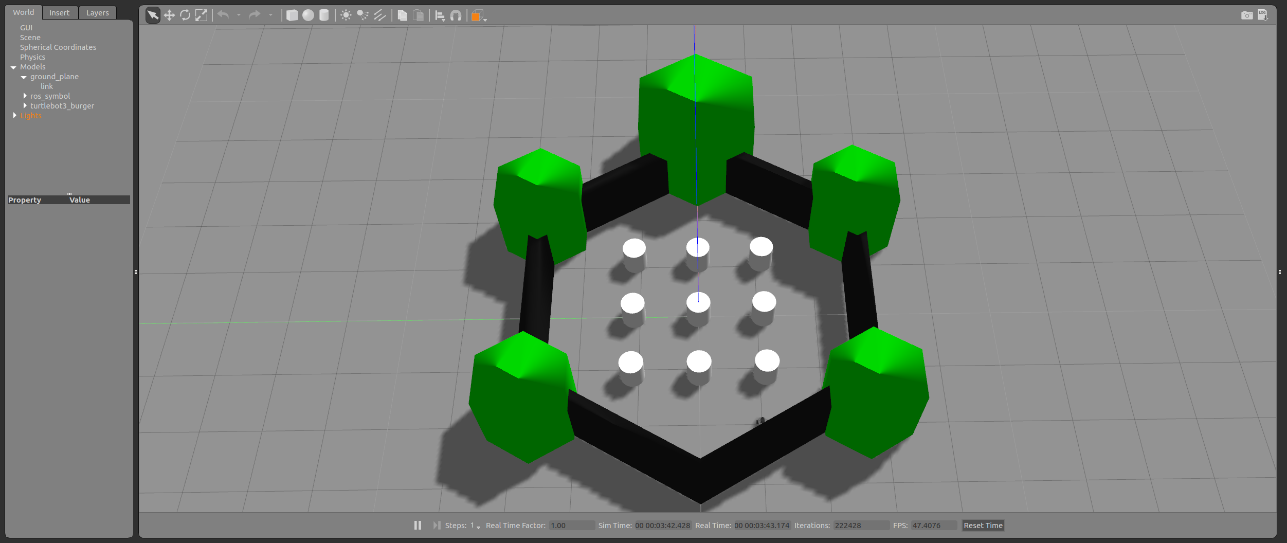


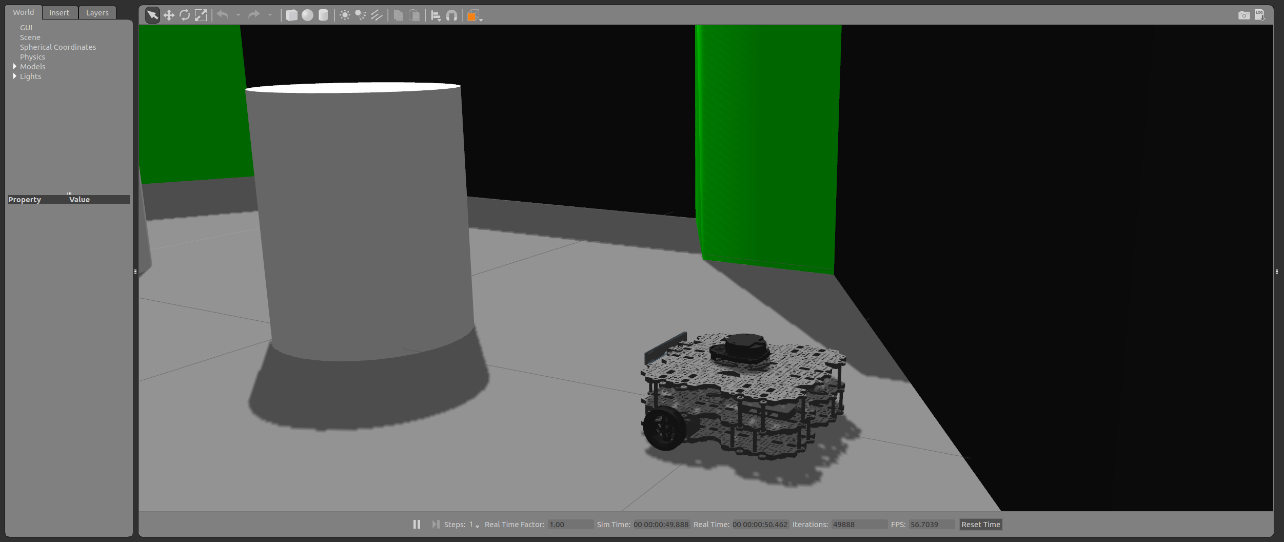
* 在Gazebo加载TurtleBot3 World

$ export TURTLEBOT3\_MODEL=${TB3\_MODEL}

## TURTLEBOT3\_MODEL有burger, waffle或waffle\_pi三种

$ roslaunch turtlebot3\_gazebo turtlebot3\_world.launch



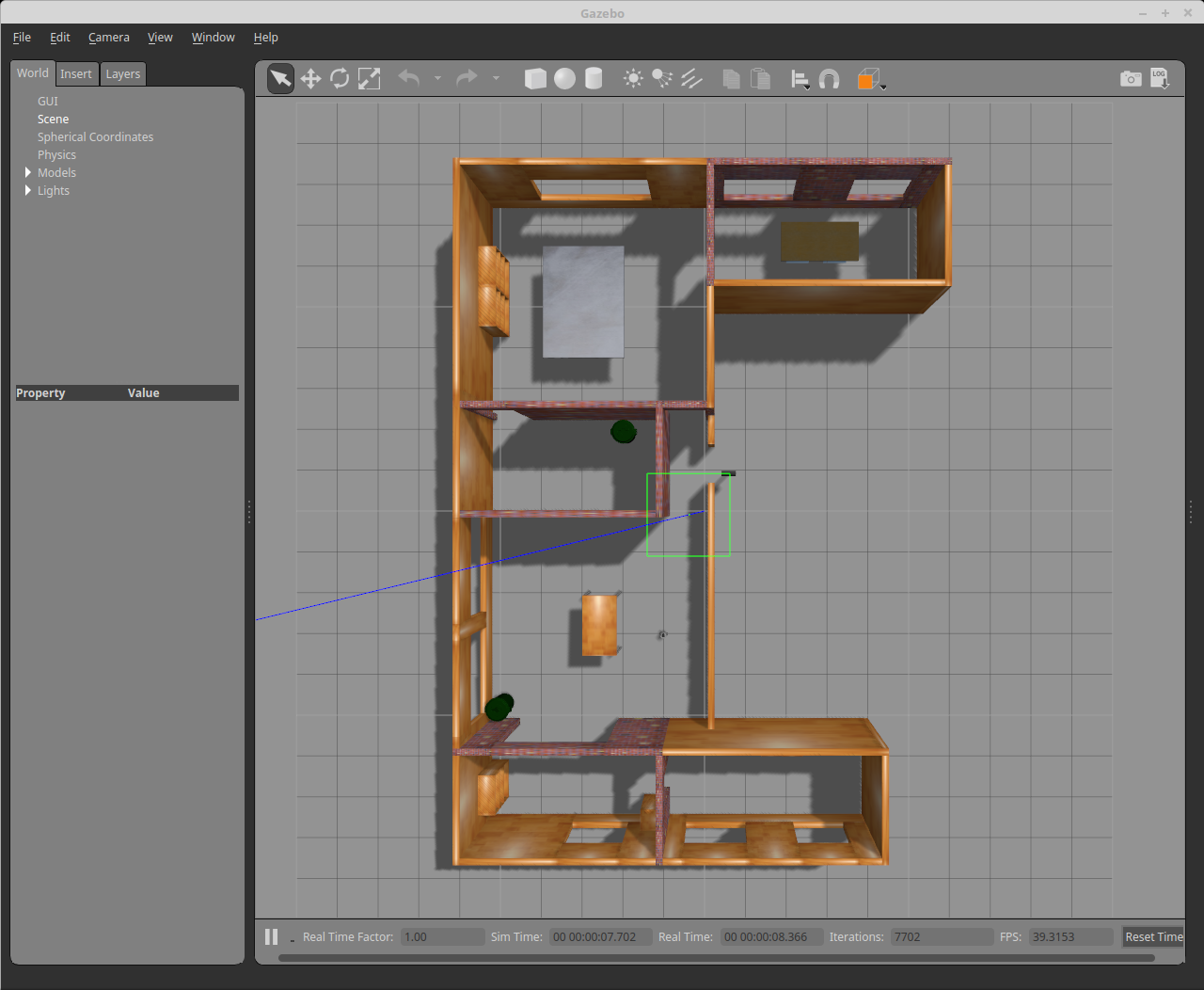


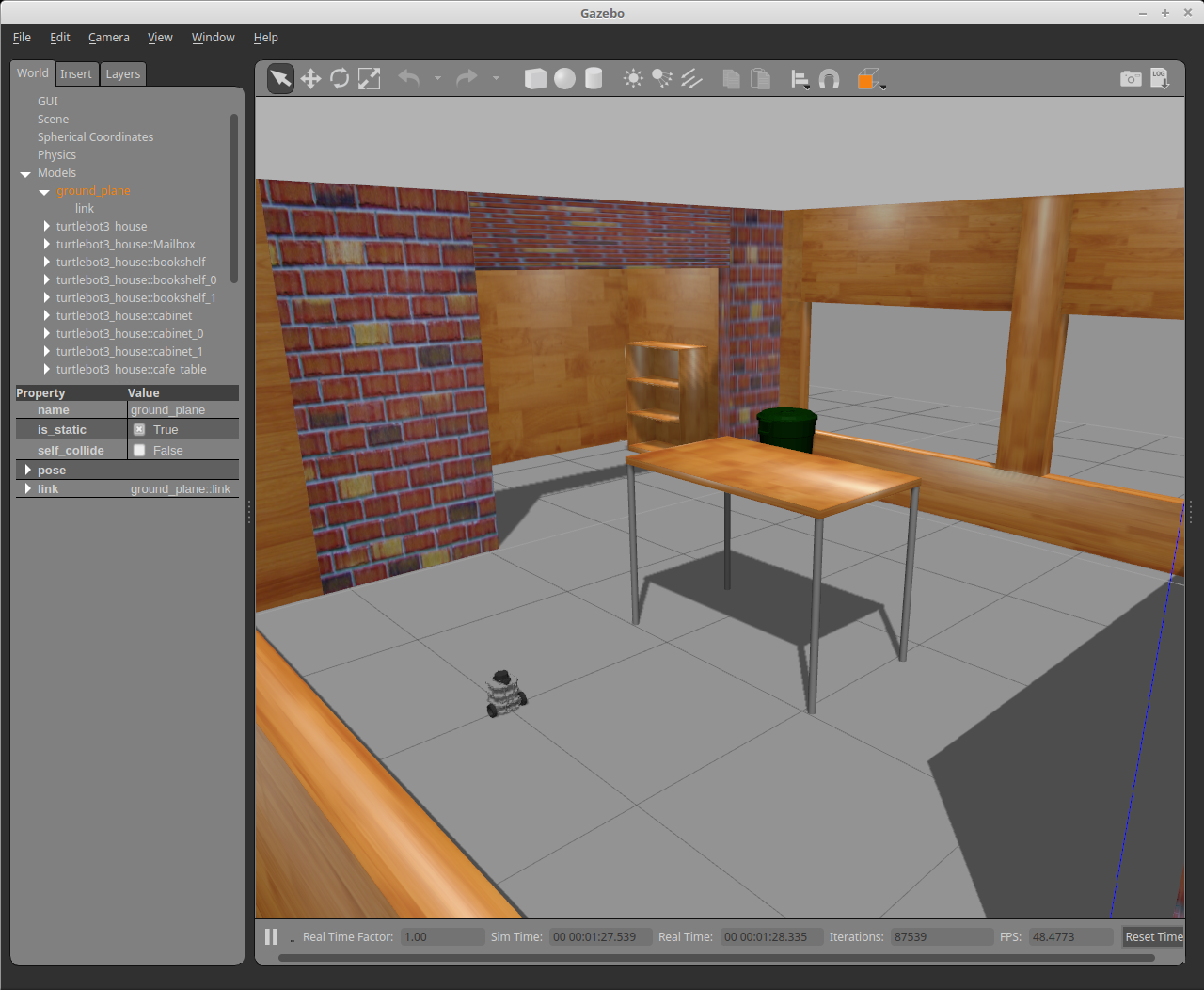
* 在Gazebo加载TurtleBot3 House

$ export TURTLEBOT3\_MODEL=${TB3\_MODEL}

## TURTLEBOT3\_MODEL有burger, waffle或waffle\_pi三种

$ roslaunch turtlebot3\_gazebo turtlebot3\_house.launch





**在Gazebo环境上控制tb3**

* 新终端，启动键盘控制

$ export TURTLEBOT3\_MODEL=${TB3\_MODEL}

## TURTLEBOT3\_MODEL有burger, waffle或waffle\_pi三种

$ roslaunch turtlebot3\_teleop turtlebot3\_teleop\_key.launch

* 新终端，加载Gazebo环境

$ export TURTLEBOT3\_MODEL=${TB3\_MODEL}

## TURTLEBOT3\_MODEL有burger, waffle或waffle\_pi三种

$ roslaunch turtlebot3\_gazebo turtlebot3\_world.launch

* 新终端，启动仿真

$ export TURTLEBOT3\_MODEL=${TB3\_MODEL}

## TURTLEBOT3\_MODEL有burger, waffle或waffle\_pi三种

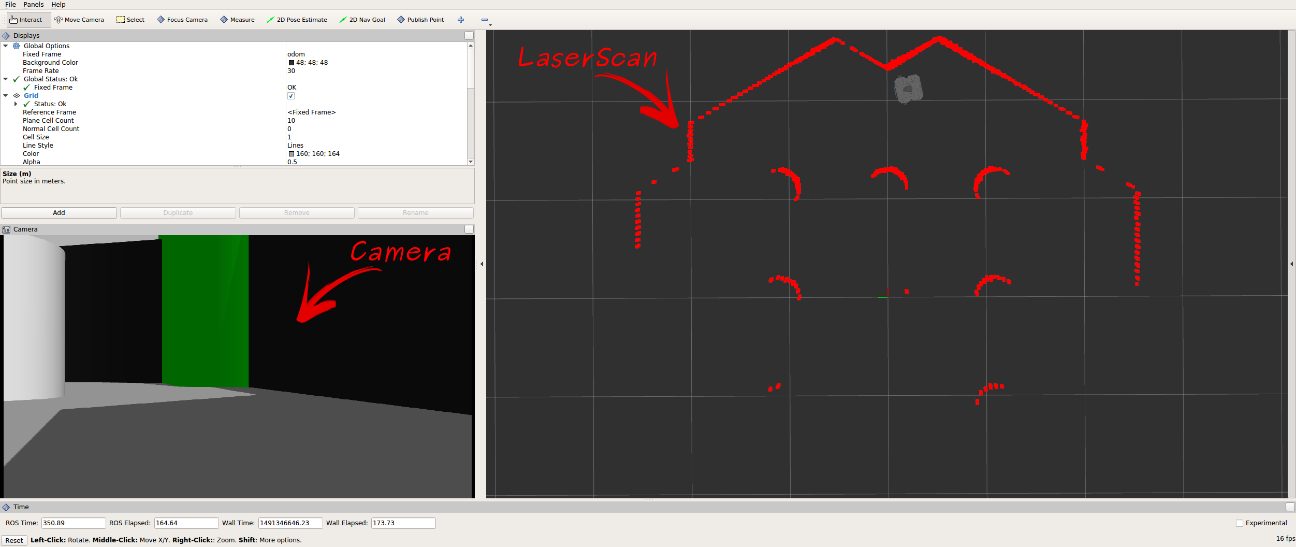
$ roslaunch turtlebot3\_gazebo turtlebot3\_simulation.launch

* 新终端，启动RViz

$ export TURTLEBOT3\_MODEL=${TB3\_MODEL}

## TURTLEBOT3\_MODEL有burger, waffle或waffle\_pi三种

$ roslaunch turtlebot3\_gazebo turtlebot3\_gazebo\_rviz.launch



**建立地图**

* 新终端，加载Gazebo环境

$ export TURTLEBOT3\_MODEL=burger

## TURTLEBOT3\_MODEL有burger, waffle或waffle\_pi三种

$ roslaunch turtlebot3\_gazebo turtlebot3\_world.launch

* 新终端，启动SLAM程序

$ export TURTLEBOT3\_MODEL=burger

## TURTLEBOT3\_MODEL有burger, waffle或waffle\_pi三种

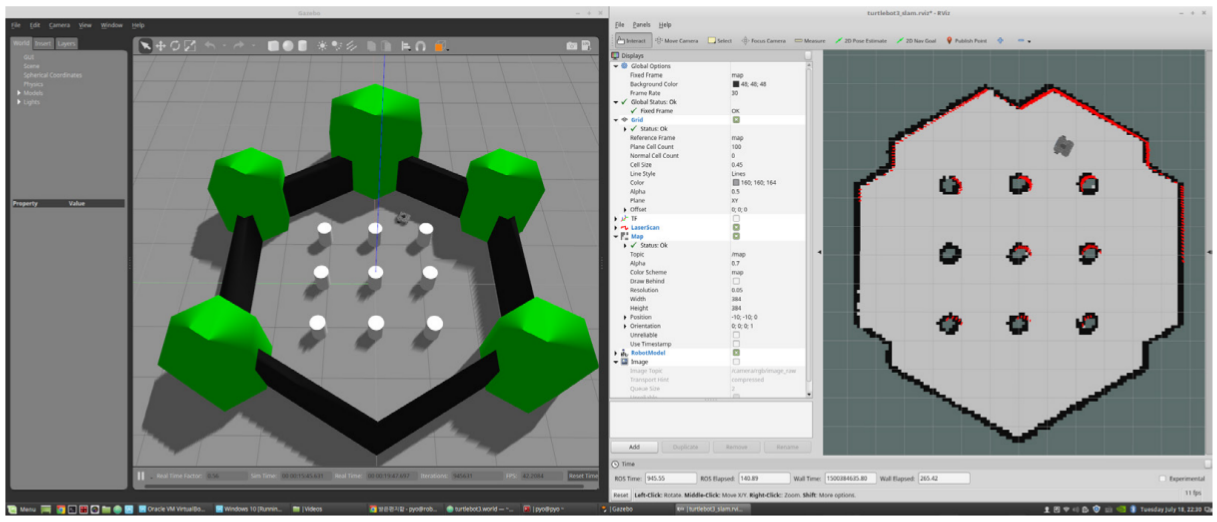
$ roslaunch turtlebot3\_slam turtlebot3\_slam.launch slam\_methods:=gmapping

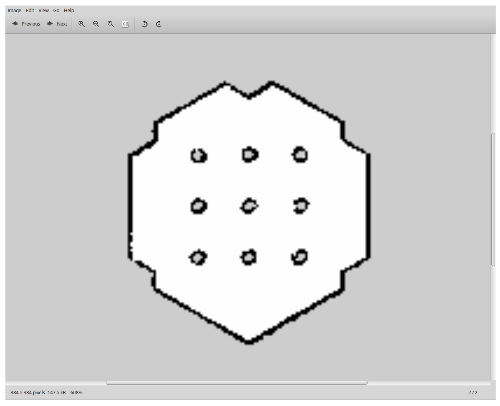
* 新终端，启动键盘控制程序

$ roslaunch turtlebot3\_teleop turtlebot3\_teleop\_key.launch

* 移动机器人进行建图，建图完成后保存地图
* 新终端，保存地图

$ rosrun map\_server map\_saver -f ~/map





**机器人导航**

**操作步骤**

* 新终端，加载Gazebo环境

$ export TURTLEBOT3\_MODEL=burger

## TURTLEBOT3\_MODEL有burger, waffle或waffle\_pi三种

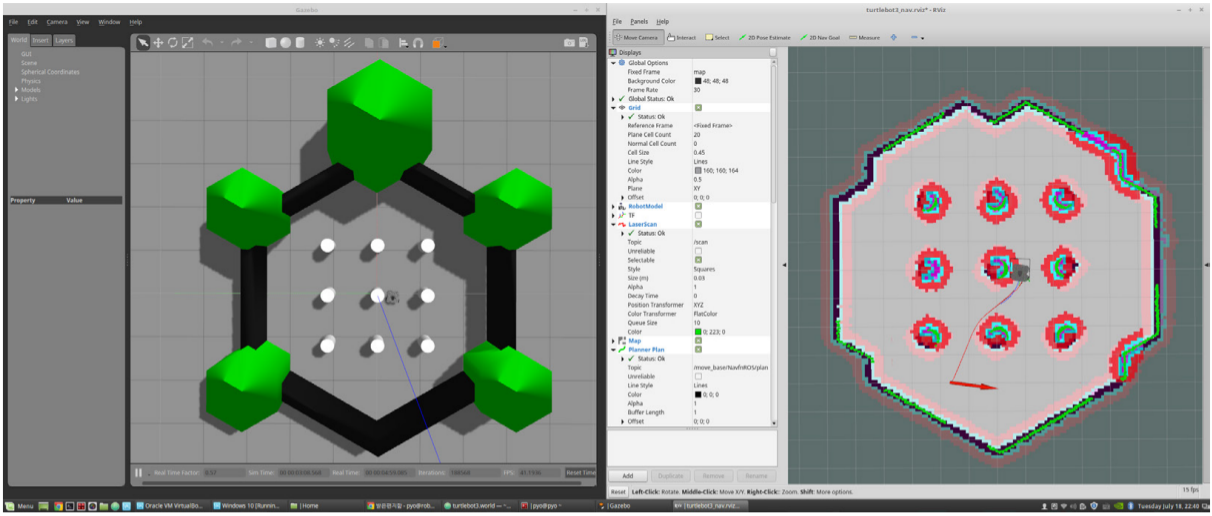
$ roslaunch turtlebot3\_gazebo turtlebot3\_world.launch

* 新终端，启动导航程序

$ export TURTLEBOT3\_MODEL=burger

## TURTLEBOT3\_MODEL有burger, waffle或waffle\_pi三种

$ roslaunch turtlebot3\_navigation turtlebot3\_navigation.launch map\_file:=$HOME/map.yaml



你启动turtlebot3的时候设置了环境变量，只是在那个终端有效，所以我们要重新在这个终端设置一次，

我这里用的是turtlebot3的 burger模型，所以在新的终端里输入：

**export TURTLEBOT3\_MODEL=burger**

出现Invalid <arg> tag: environment variable 'TURTLEBOT3\_MODEL' is not set.