1.1gmapping仿真环境

接下来我们看看基于gmapping仿真建图的效果是什么样的，所有源码已经分享了，这里面有几个功能包下载下来，然后在工作空间下对代码进行编译。

在启动了roscore节点之后

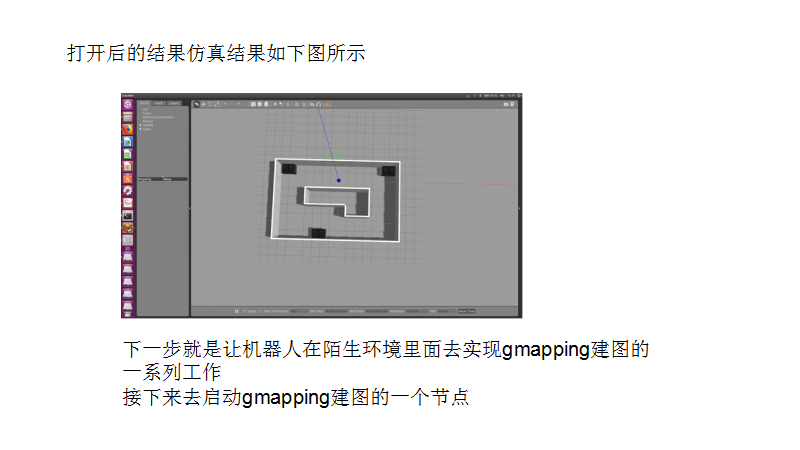


第一步使用命令启动仿真环境

$roslaunch mbot\_gazebo mbot\_laser\_nav\_gazebo.launch



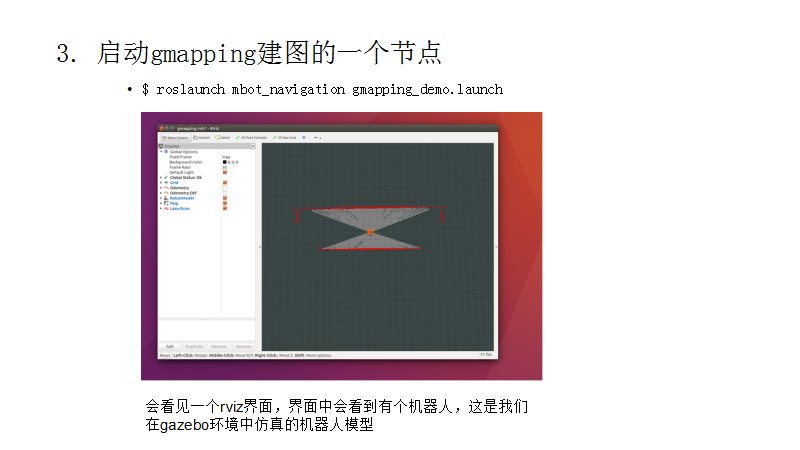
仿真环境已经启动成功了。可以看到一个机器人，我们让机器人在陌生环境里面去实现gmapping建图的工作



1.2gmapping建图

接下来去启动gmapping建图的一个节点,

$ roslaunch mbot\_navigation gmapping\_demo.launch



会看见一个rviz界面，界面中会看到有个机器人，这是我们在gazebo环境中仿真的机器人模型。机器人周围会有很多红色的点，这些红色的点就是上面我们安置的激光雷达，他所检测到的周围环境的深度信息。跟gazebo环境对比，这些红色的点就是墙体，同时在rviz中可以看到一些浅灰色的信息，这些就是目前机器人就是已经创建完成的地图。

将激光雷达关掉

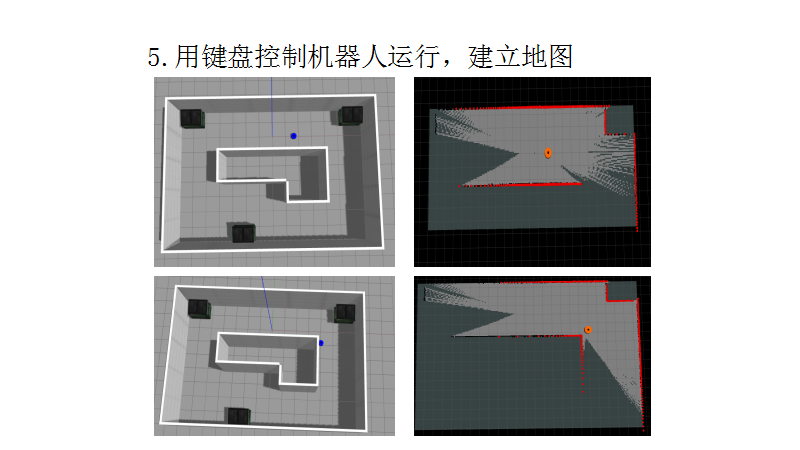
可以看到最边缘是黑色的点，这些灰色的点代表地图建立完成的障碍物，灰色的区域空间机器人是可以运动的。跟实际的仿真环境是相似的。接下来让机器人在仿真的环境中做运动，在运动当中会把周围的环境信息建立出来，建立出一张完整的二维三格地图。

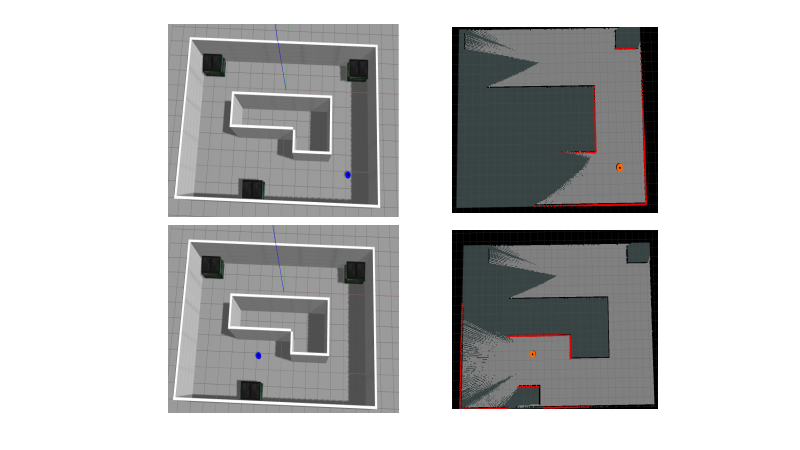
第三步启动键盘节点，这里我们首先使用键盘控制来人为的控制机器人在键盘中做运动。

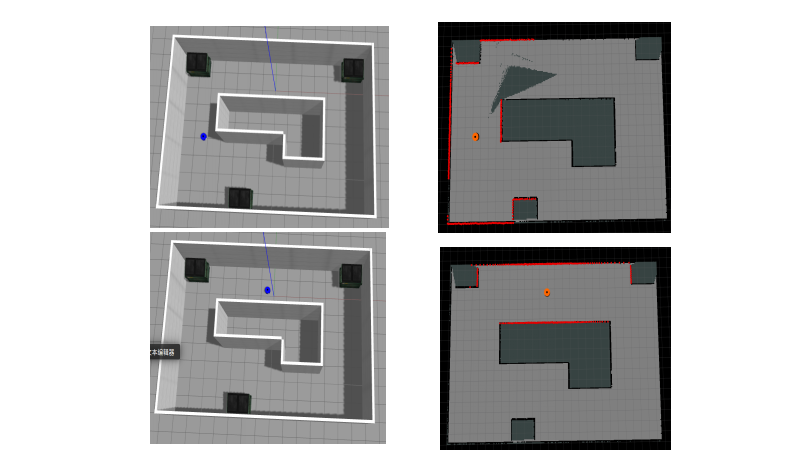
输入$ roslaunch mbot\_teleop mbot\_teleop.launch

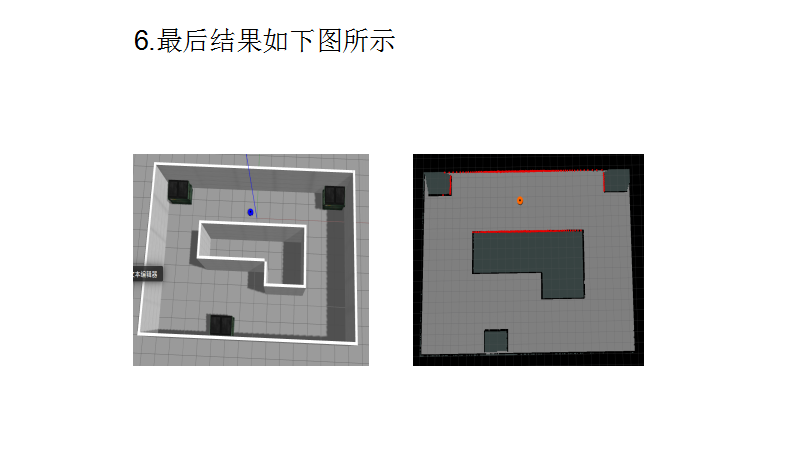


摆好界面用键盘控制机器人运动。







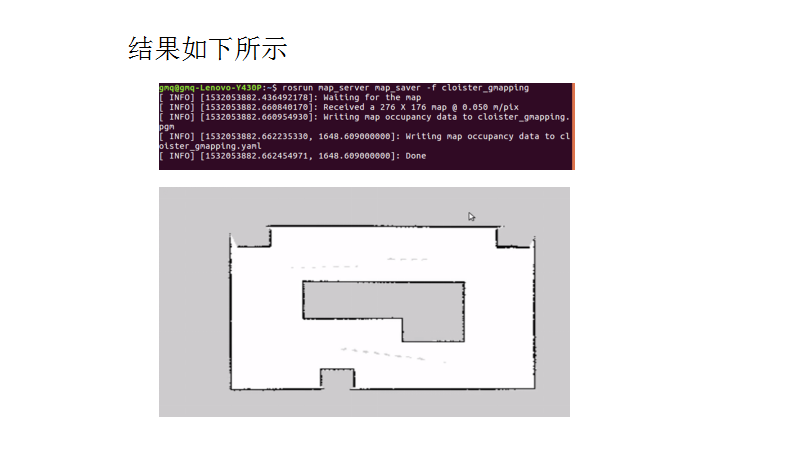


1.3保存图片

激光雷达检测范围比较远，可以看出封闭性还是不错的，建立的信息和之前的地图也是重合的。

大家可以使用这样一句命令把地图保存下来。





1.1Hector仿真环境

接下来我们看看基于gmapping仿真建图的效果是什么样的，所有源码已经分享了，这里面有几个功能包下载下来，然后在工作空间下对代码进行编译。

在启动了roscore节点之后

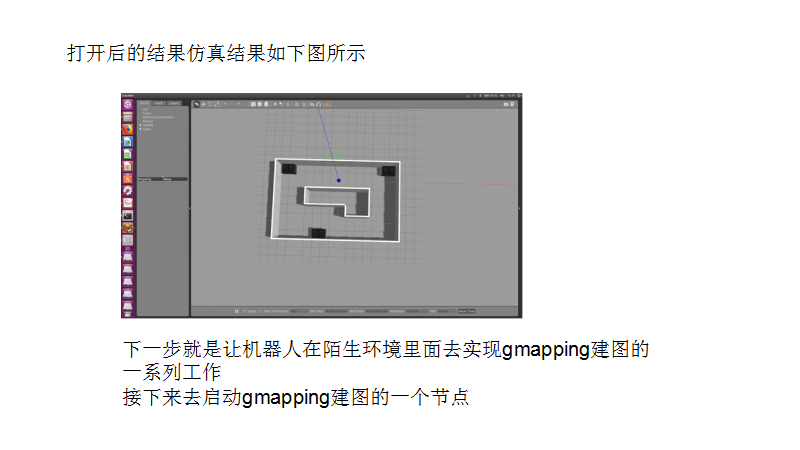


第一步使用命令启动仿真环境

$roslaunch mbot\_gazebo mbot\_laser\_nav\_gazebo.launch



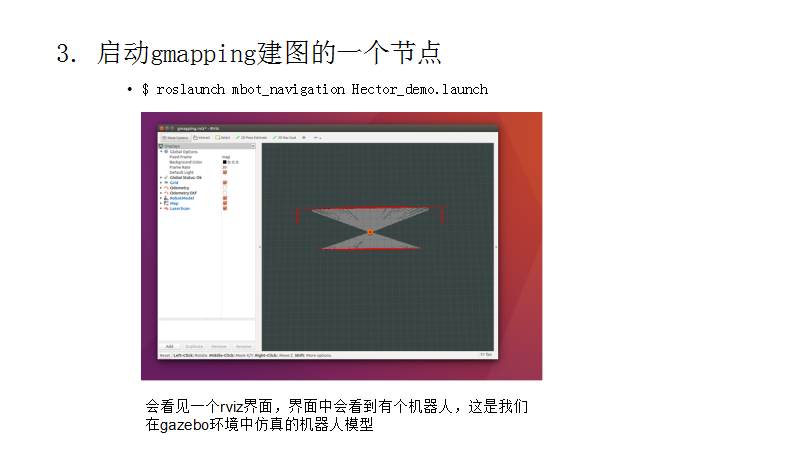
仿真环境已经启动成功了。可以看到一个机器人，我们让机器人在陌生环境里面去实现gmapping建图的工作



1.2gmapping建图

接下来去启动gmapping建图的一个节点,

$ roslaunch mbot\_navigation hector\_demo.launch



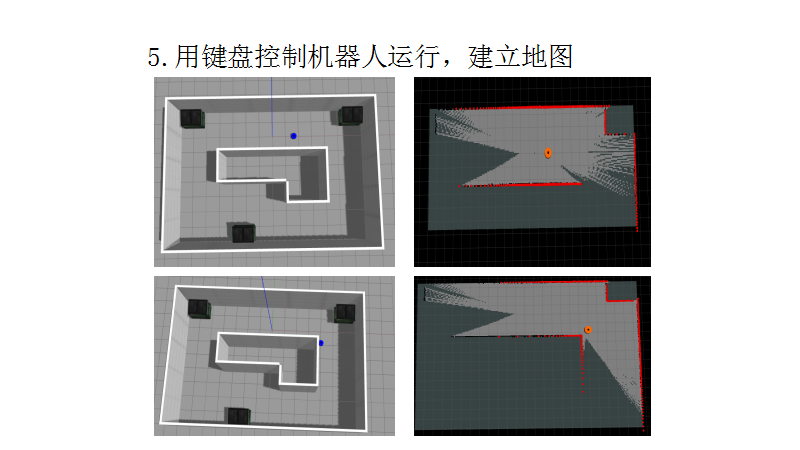
接下来让机器人在仿真的环境中做运动，在运动当中会把周围的环境信息建立出来，建立出一张完整的二维三格地图。

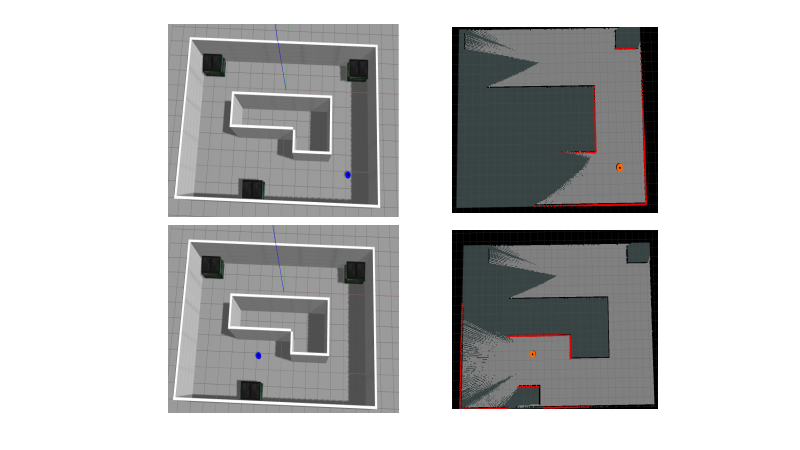
第三步启动键盘节点，这里我们首先使用键盘控制来人为的控制机器人在键盘中做运动。

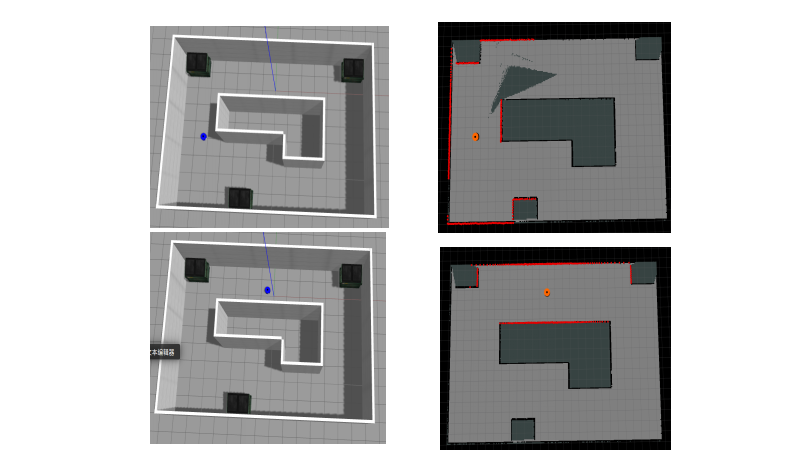
输入$ roslaunch mbot\_teleop mbot\_teleop.launch

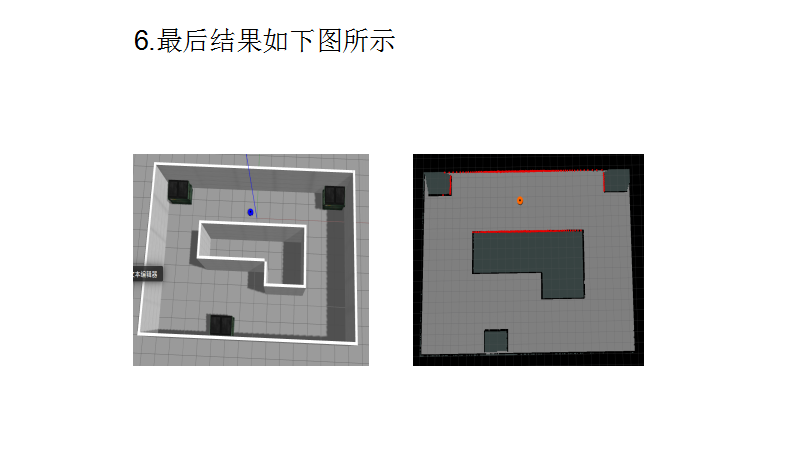


摆好界面用键盘控制机器人运动。







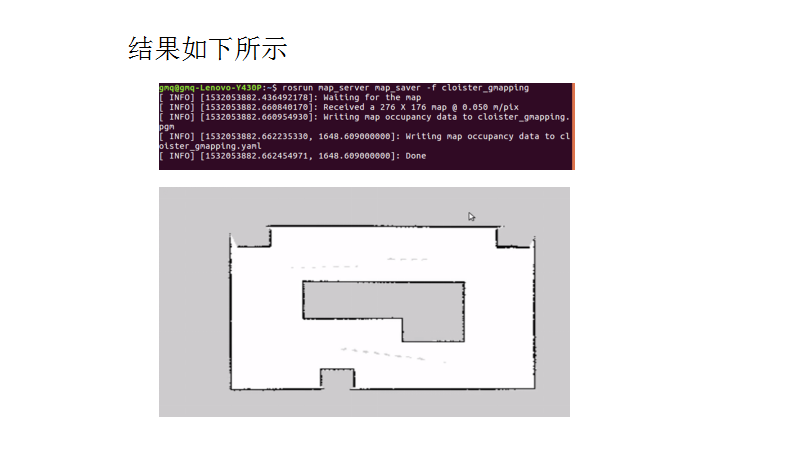


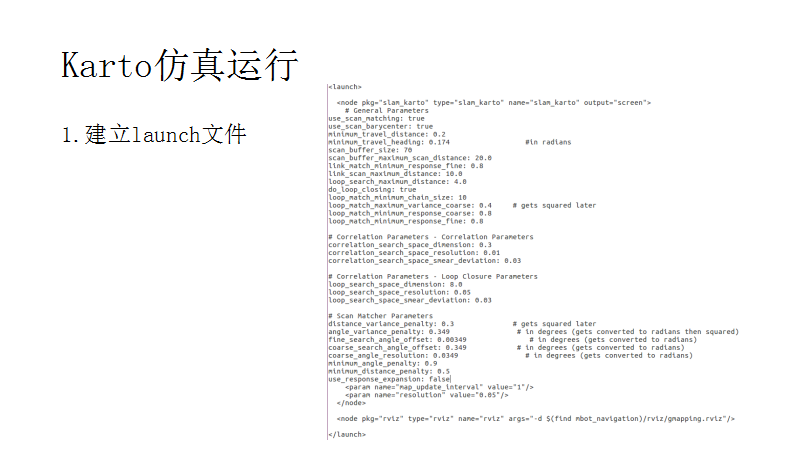
1.3保存图片

激光雷达检测范围比较远，可以看出封闭性还是不错的，建立的信息和之前的地图也是重合的。

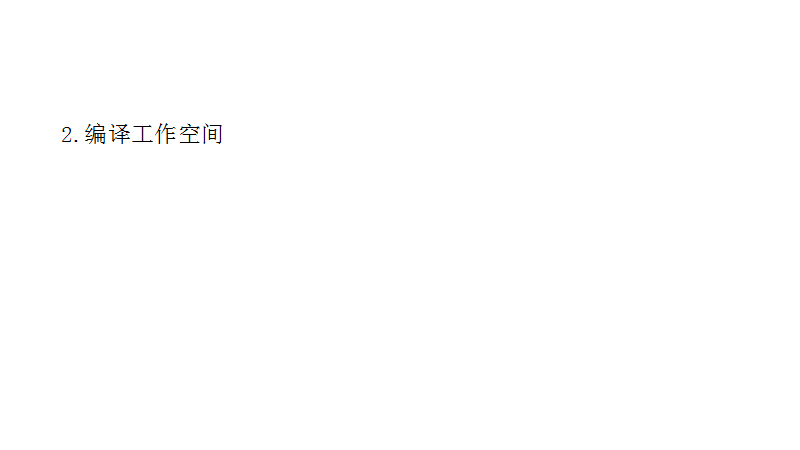
大家可以使用这样一句命令把地图保存下来。







由于之前我们已经在xbot仿真环境中运行过Karto程序包，所以我们的ROS系统中已经安装了Karto功能包，以及相关依赖，不需要我们再去安装，直接编写相应的launch文件就可以了。首先新建一个文档，并把后缀改为.launch文件，里面可以直接将karto功能包中的launch文件内容复制粘贴过来，可以不修改里面的参数，也可以尝试修改一下，看看效果。这个launch文件中最后一行是直接打开已经配置好的rviz文件，我们可以省略打开的rviz界面，当然你也可以不写入launch文件，单独打开也是一样的。

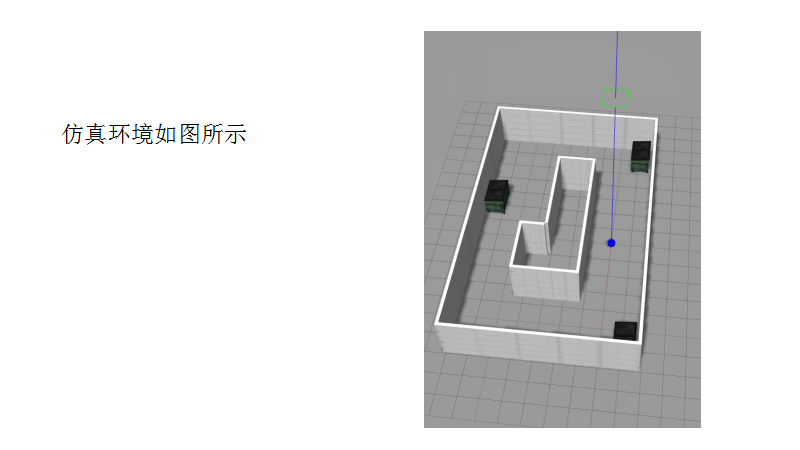


紧接着把这个launch文件放到工作空间目录下，按照约定俗成的规矩，我们放入launch文件包中的mbot\_navigation包中，方便寻找。

紧接着启动roscore，这是每一个ros运行程序都必须的一步。



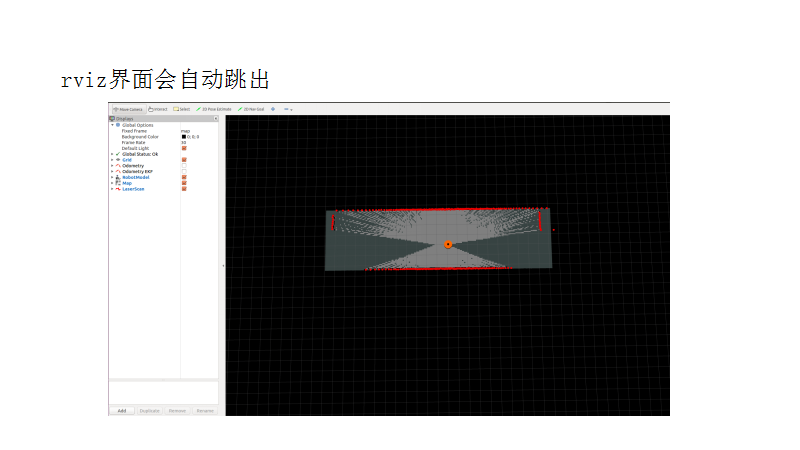
然后就是打开仿真环境，第一次在打开的时候需要等比较长的时间，等待电脑将仿真环境加载出来，第二次打开就会快很多。但是经常在打开仿真环境的时候会出现gazebo程序崩溃的情况，多试几次就可以。



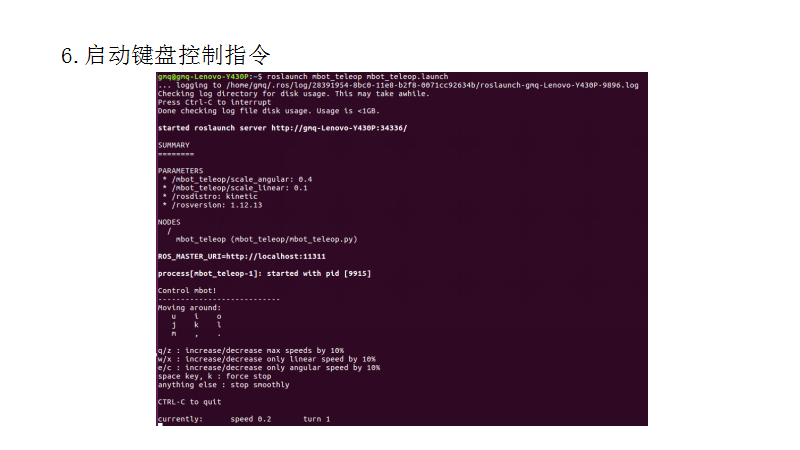
仿真环境像图上那样。



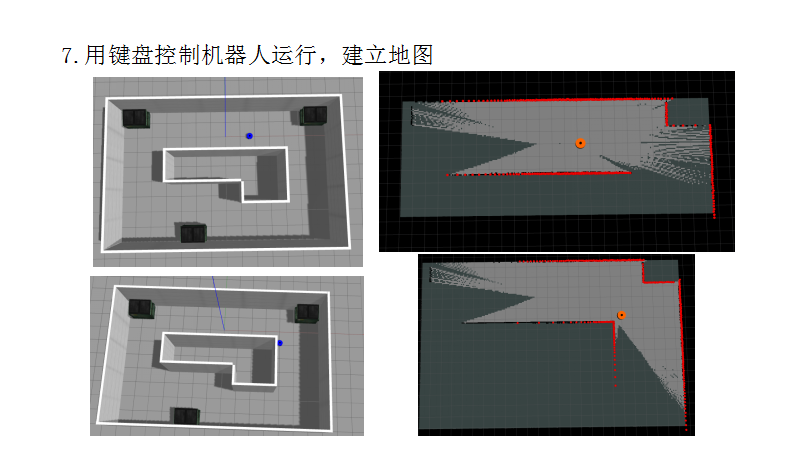
下一步就是启动我们自己编写的karto文件，根据自己launch文件放置的位置不同自己输入相对应的命令。运行后会出现如下结果，同时rviz界面会自动跳出。

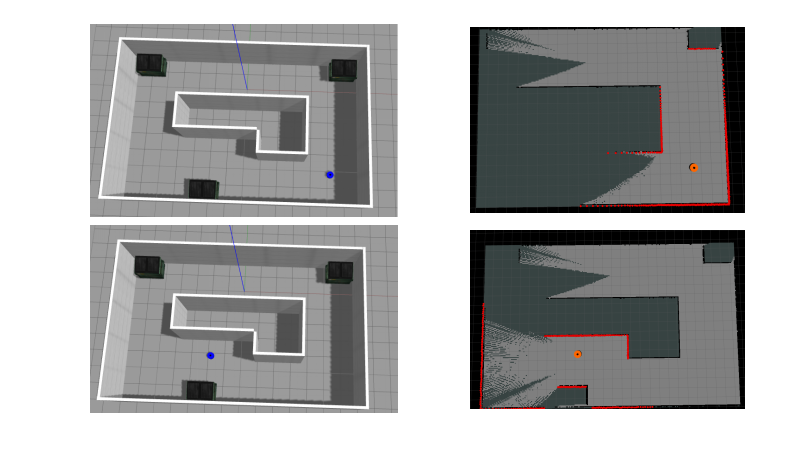


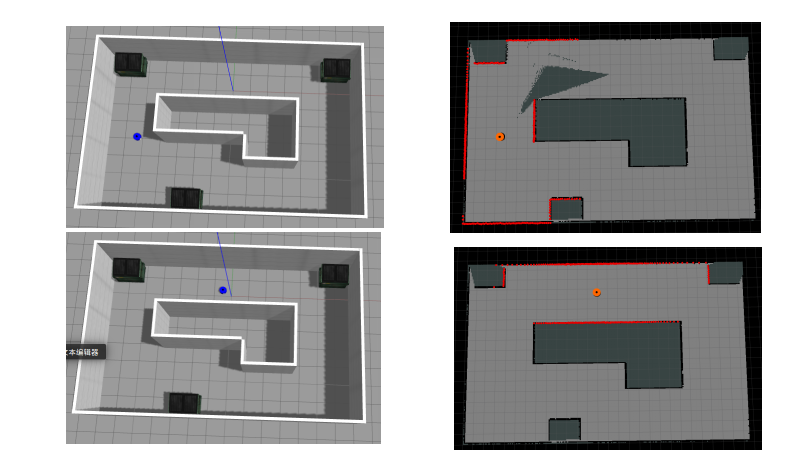
rviz界面就是这样，注意一定要有激光雷达的红色点阵，不然就是rviz配置有问题。



为了建立地图，我们需要控制机器人四处移动，输入控制机器人移动的指令然后操作机器人绕环境走一圈，把地图都探索完。









在地图探索完之后，保存地图，我们可以看到如图所示的地图。