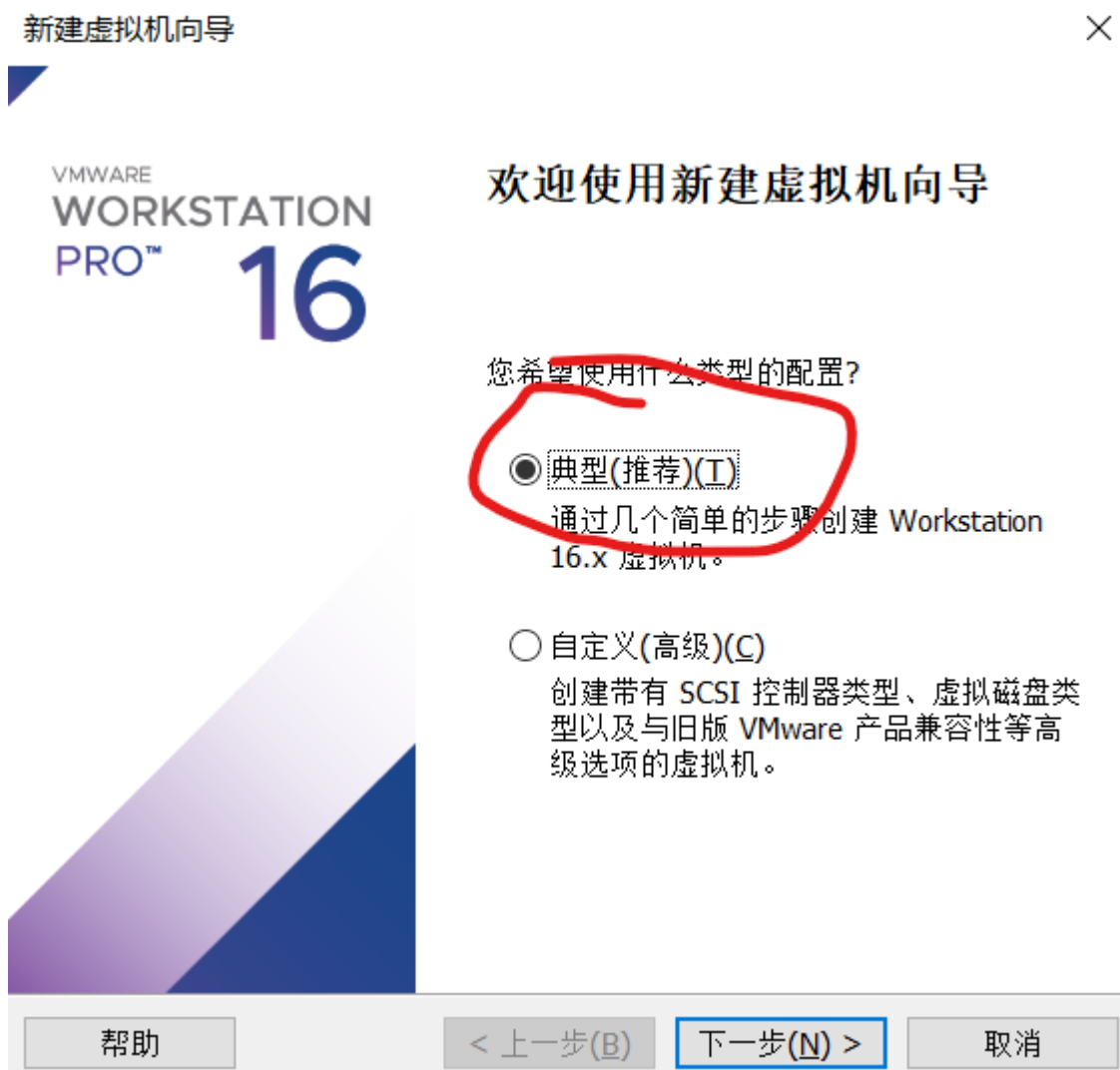# 1.安装ubuntu和ros

本教程的Ubuntu操作系统是16.04，后面所有的程序配置都是在这个版本的Linux上运行的。

本教程以虚拟机为运行环境，一步一步配置Baxter仿真环境。
大家也可以不用虚拟机，自己去安装一个Linux系统，配置操作都是一样的，而且在仿真Baxter的时候更快速，更流畅。

1.1 虚拟机的创建

首先利用Vmware创建虚拟机，点击典型

然后，选择下载好的Ubuntu16.04的iso镜像

新建虚拟机向导   ✕

安装客户机操作系统
    虚拟机如同物理机，需要操作系统。您将如何安装客户机操作系统?

安装来源:

○ 安装程序光盘(D):

    无可用驱动器

◉ 安装程序光盘映像文件(iso)(M):

    D:\单片机\树莓派\ubuntu-16.04.7-desktop-amd64.iso ∨   浏览(R)...

    ⓘ 已检测到 Ubuntu 64 位 16.04.7。
        该操作系统将使用简易安装。(这是什么?)

○ 稍后安装操作系统(S)。

    创建的虚拟机将包含一个空白硬盘。

    帮助   < 上一步(B)   下一步(N) >   取消

然后为虚拟机设定用户名和密码，选择虚拟机的位置，再为其分配存储空间

新建虚拟机向导   ✕

指定磁盘容量
　　磁盘大小为多少?

虚拟机的硬盘作为一个或多个文件存储在主机的物理磁盘中。这些文件最初很小，
随着您向虚拟机中添加应用程序、文件和数据而逐渐变大。

最大磁盘大小 (GB)(S):   20.0 ▲▼

针对 Ubuntu 64 位 的建议大小: 20 GB

◉ 将虚拟磁盘存储为单个文件(O)
◯ 将虚拟磁盘拆分成多个文件(M)
　　拆分磁盘后，可以更轻松地在计算机之间移动虚拟机，但可能会降低大容量磁
　　盘的性能。

| 帮助 | | < 上一步(B) | 下一步(N) > | 取消 |

至此，Ubuntu16.04的虚拟机环境就创建完成了

1.2 安装ros系统

一般来说，安装ros是非常耗时间且痛苦的步骤。幸运的是，有人制作了一个自动化安装ros的脚本:
对于Ubuntu16.04，我们需要安装ros kinetic版本
我们只需要在控制台输入以下命令:

```
wget http://fishros.com/install -O fishros && sudo bash fishros
```

接下来按照提示一步一步操作就好了

接下来，稍微等待一会，你可以去休息以下喝杯咖啡~，直到安装结束

下面我们来测试一下ros是否安装成功了：
我们将会运行一个入门的ros例程来检验是否安装成功
首先，在控制台终端输入(这个roscore就一直留着别关闭，后面仿真的时候要开着的)

```
roscore
```

出现如下字样：

```
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ubuntu:32849/
ros_comm version 1.12.17


SUMMARY
========

PARAMETERS
 * /rosdistro: kinetic
 * /rosversion: 1.12.17

NODES

auto-starting new master
process[master]: started with pid [34068]
ROS_MASTER_URI=http://ubuntu:11311/

setting /run_id to ff992566-244f-11ed-8183-000c294ce1a7
process[rosout-1]: started with pid [34081]
started core service [/rosout]
```
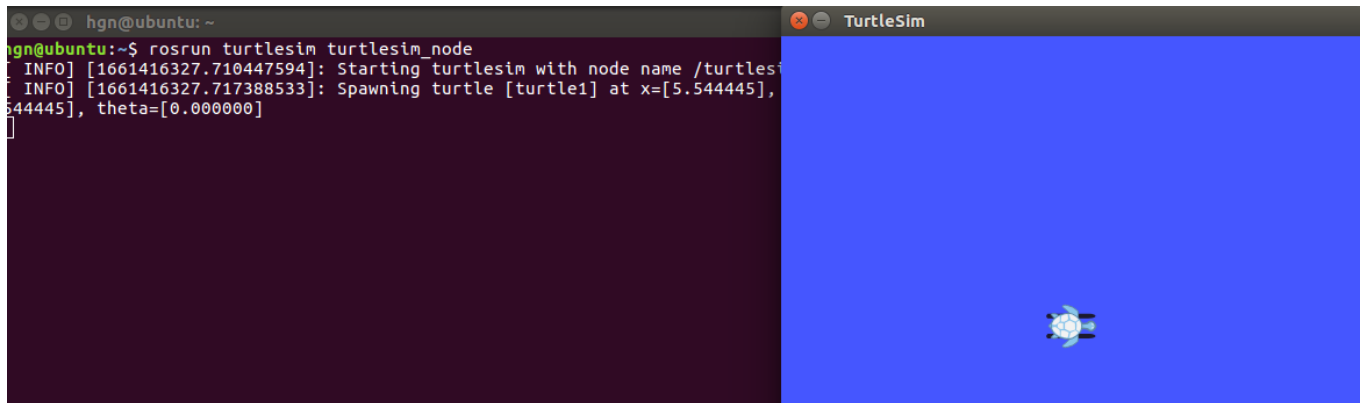
然后，新建一个终端，输入

```
rosrun turtlesim turtlesim_node
```

会出现一个小王八

再新建一个终端

```
rosrun turtlesim turtle_teleop_key
```

输入之后，你可以用键盘控制王八的行走
最后，再新开一个终端，输入：

```
rosrun rqt_graph rqt_graph
```

可以看到ros的图形化界面，展示了ros节点之间的关系
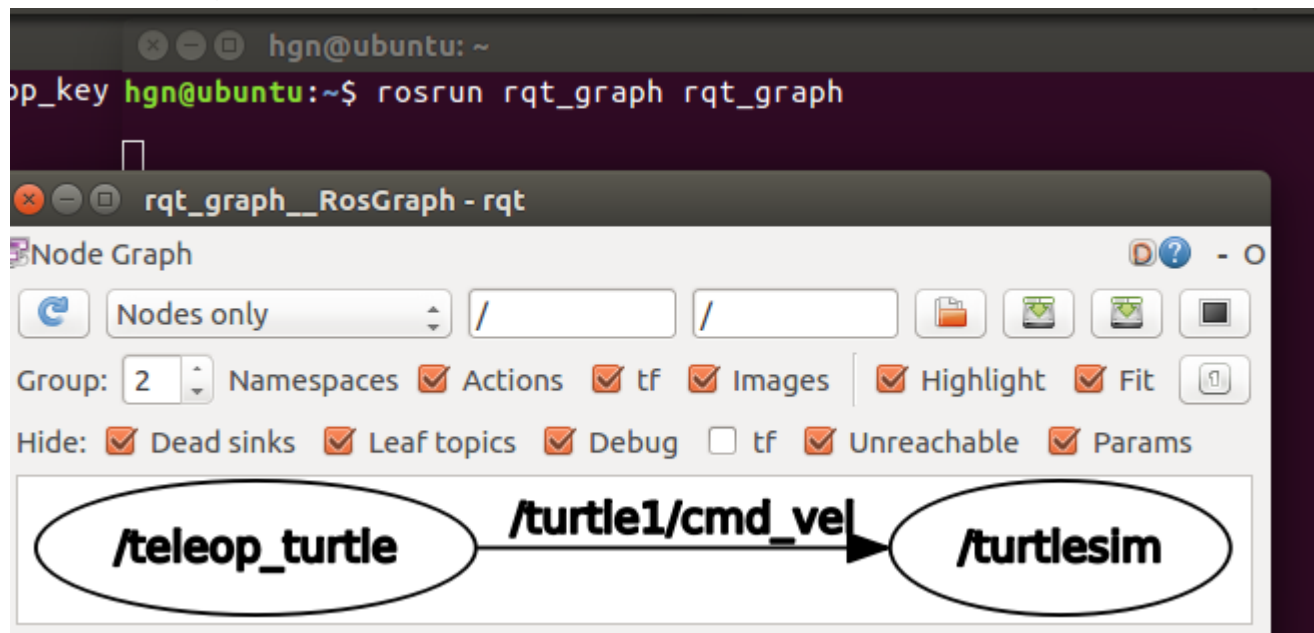
如果你以上三步都成功运行，恭喜你安装ros成功了！
至此，我们已经配置好了ros系统！

---

2. 功能包和**IDE**的安装

在配置好ros系统之后，我们还需要安装一些包，一些Baxter仿真所依赖的功能包

2.1 依赖功能包的安装

安装moveit

```
sudo apt-get install ros-kinetic-moveit
```

安装完成后，在终端输入查看是否安装成功

```
moveit_version
```

安装如下包，都是baxter仿真必需的包(我也不知道为什么要安装以下这些包，都是报错一个个试出来的)

```
sudo apt-get install ros-kinetic-moveit-ros-planning-interface
```

```
sudo apt-get install ros-kinetic-moveit-ros-perception
```

```
sudo apt-get install ros-kinetic-moveit-msgs
```

```
sudo apt-get install ros-kinetic-gazebo-ros-control
```

```
sudo apt-get install ros-kinetic-effort-controllers
```

```
sudo apt-get install ros-kinetic-manipulation-msgs
```

```
sudo apt-get install ros-kinetic-move-base-msgs
```

```
sudo apt-get install ros-kinetic-ros-control ros-kinetic-ros-controllers
```

2.2 安装ROS IDE —— ROBOWARE STUDIO

RoboWare Studio是一个ROS集成开发环境.使ROS开发更加直观、简单，并且易于操作.可进行
ROS工作空间及包的管理，代码编辑、构建及调试。
本文以安装Roboware为示例进行安装。
roboware下载链接：

https://drive.google.com/file/d/1I6HBV6B3vOqf8lb4JJ7wcO2HptVHlWoN/view?usp=sharing

Ubuntu Software

roboware-studio

roboware-studio
IDE for ROS(Robot Operating System) developers.

Install

RoboWare Studio is an IDE environment for ROS development. It makes the ROS development visual, simple
and manageable. It provides ROS workspace management, code editing, building and debugging. It is an IDE,
but it will not be an IDE only. See http://www.roboware.me/ Powered by Jinan Tony Robotics Co., Ltd.

Website

Details

Version   1.0.0-1505955059
Source   Unknown
Size   355.2 MB

进入你的安装包所在的文件夹里，打开终端，输入命令进行解压安装：

```
sudo dpkg -i + roboware-studio_1.0.0-1505955059_amd64.deb
```

之后会弹出来如下这个界面，利用键盘方向键选中Ok(即有一个红色框在Ok上)，然后按回车键。
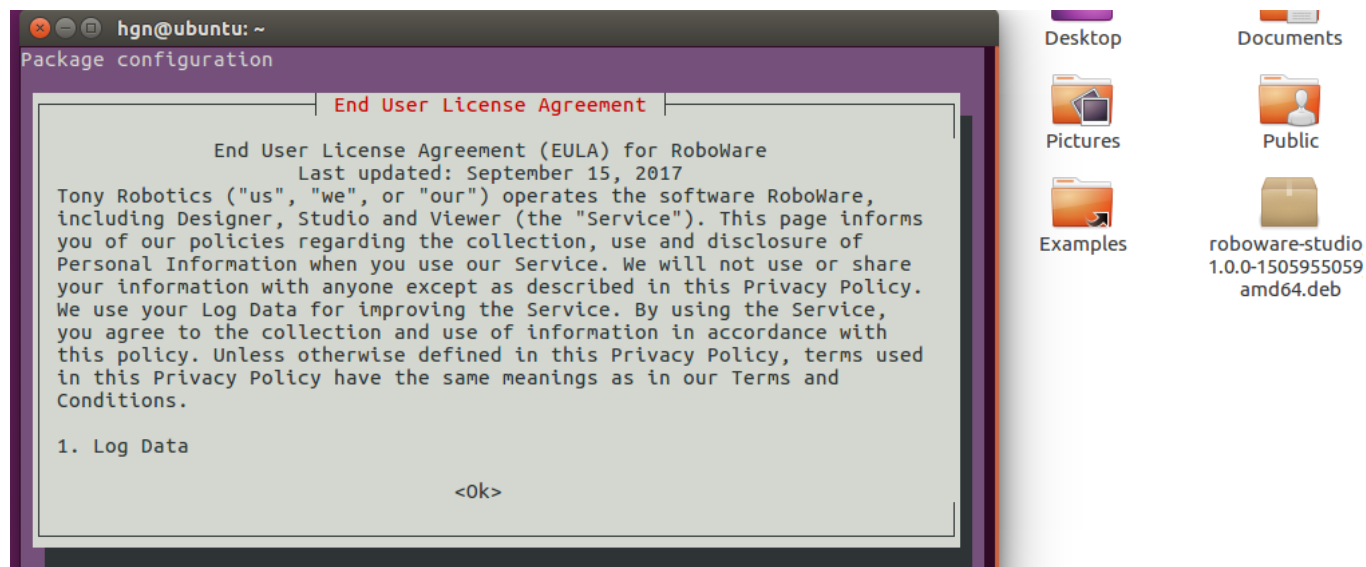
之后会出现下面这个页面，还是选中yes，按回车键
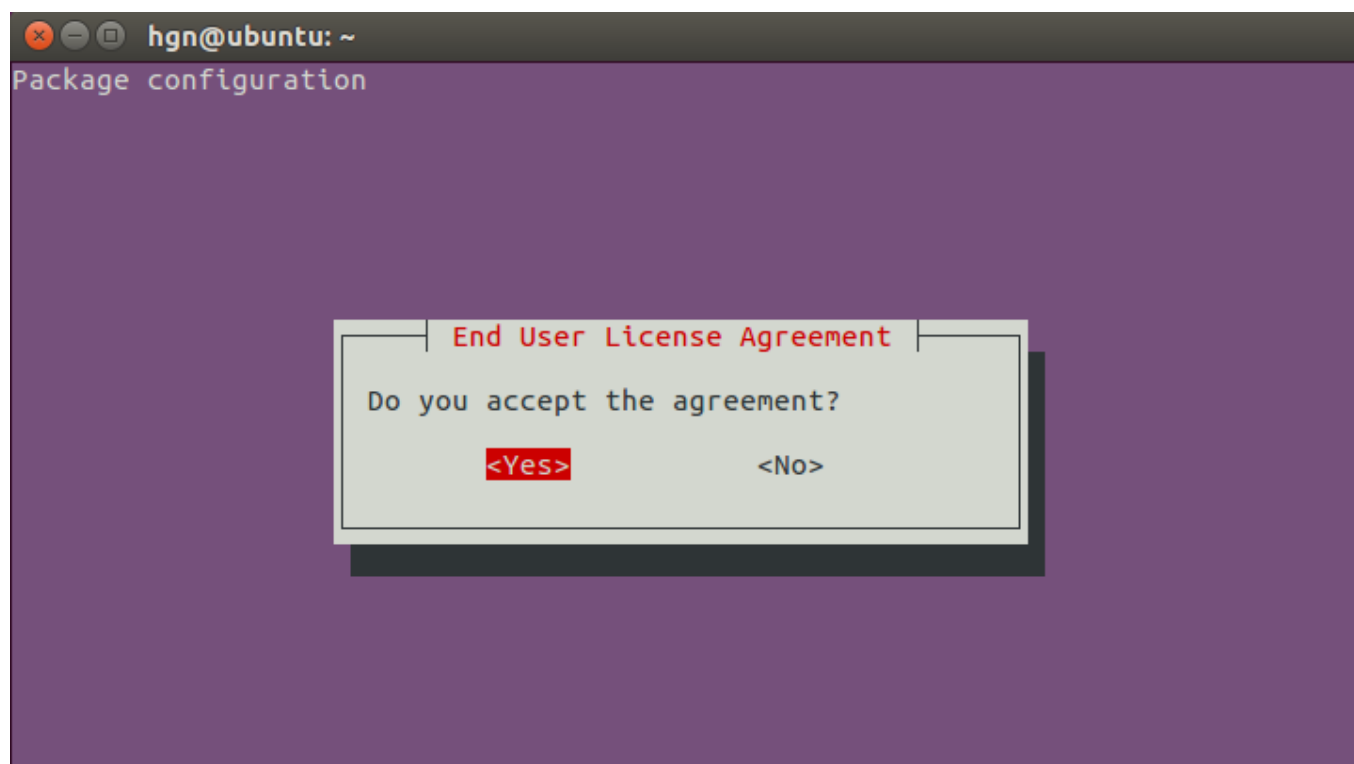
稍微等一会，出现下图所示便是安装成功了

```
hgn@ubuntu:~$ sudo dpkg -i roboware-studio_1.0.0-1505955059_amd64.deb
[sudo] password for hgn:
Selecting previously unselected package roboware-studio.
(Reading database ... 253154 files and directories currently installed.)
Preparing to unpack roboware-studio_1.0.0-1505955059_amd64.deb ...
Unpacking roboware-studio (1.0.0-1505955059) ...
Setting up roboware-studio (1.0.0-1505955059) ...
Processing triggers for gnome-menus (3.13.3-6ubuntu3.1) ...
Processing triggers for desktop-file-utils (0.22-1ubuntu5.2) ...
Processing triggers for bamfdaemon (0.5.3~bzr0+16.04.20180209-0ubuntu1) ...
Rebuilding /usr/share/applications/bamf-2.index...
Processing triggers for mime-support (3.59ubuntu1) ...
hgn@ubuntu:~$
```

接下来，双击打开roboware应用

RoboWare Studio
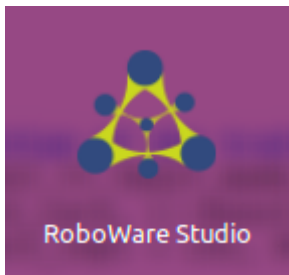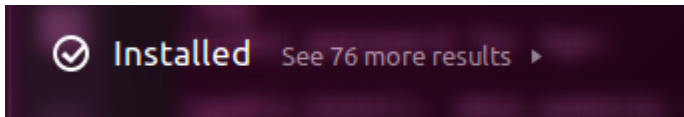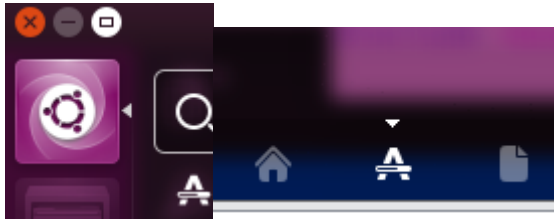
如下我们成功的打开了应用
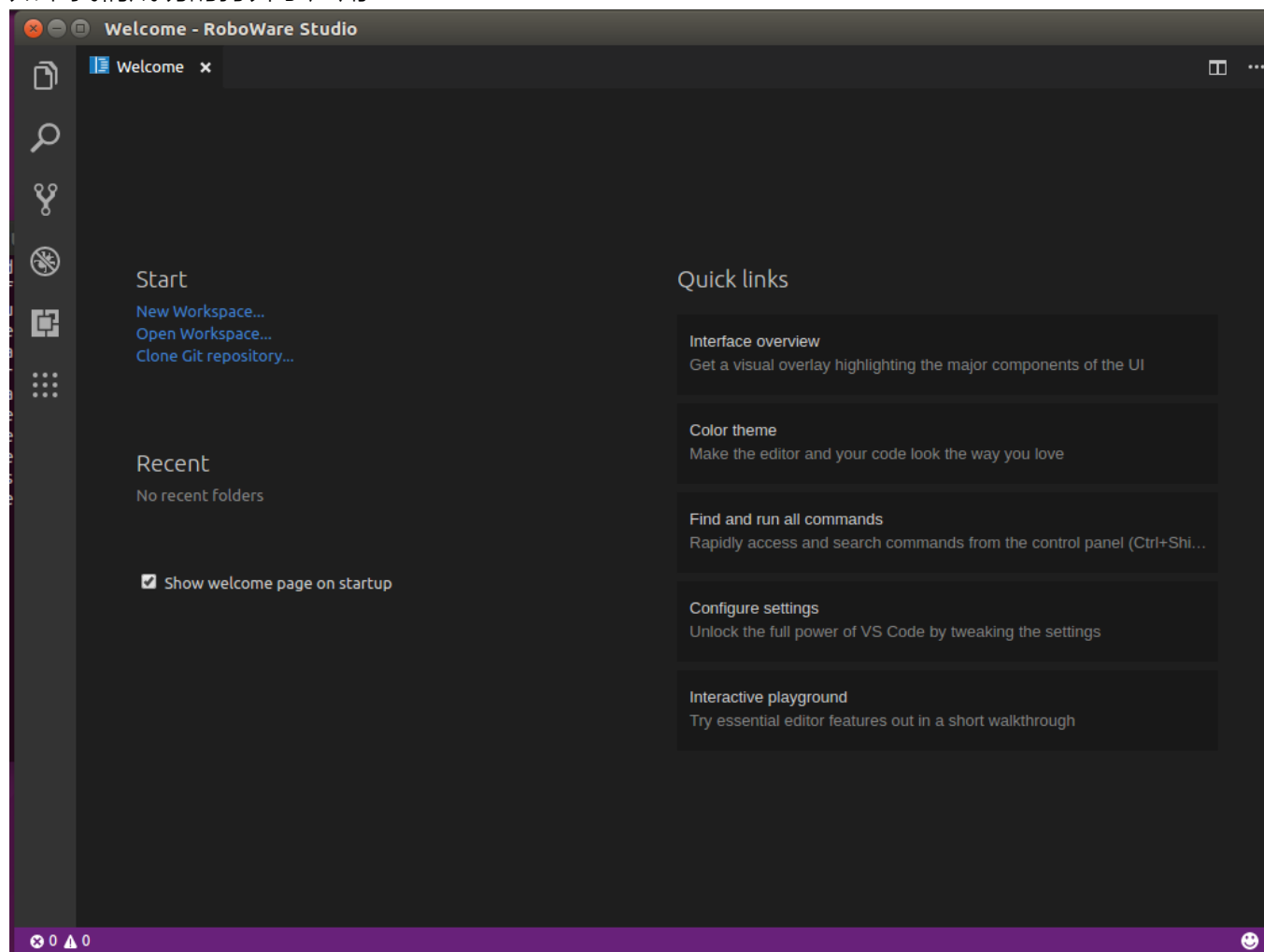
3. Baxter仿真环境的安装!

3.1 创建工作空间

在这里，我们要创建一个工作空间（这里起名叫做catkin_ws)

创建文件夹

```
mkdir -p ~/catkin_ws/src
```

移动到src文件夹下

```
cd ~/catkin_ws/src
```

初始化工作空间

```
catkin_init_workspace
```

```
cd ~/catkin_ws/
```

编译工作空间，即使这个工作空间什么都没有，我们还是可以编译

```
catkin_make
```

```
hgn@ubuntu:~$ mkdir -p ~/catkin_ws/src
hgn@ubuntu:~$ cd ~/catkin_ws/src
hgn@ubuntu:~/catkin_ws/src$ catkin_init_workspace
Creating symlink "/home/hgn/catkin_ws/src/CMakeLists.txt" pointing to "/opt/ros/
kinetic/share/catkin/cmake/toplevel.cmake"
hgn@ubuntu:~/catkin_ws/src$ cd ~/catkin_ws/
hgn@ubuntu:~/catkin_ws$ catkin_make
Base path: /home/hgn/catkin_ws
Source space: /home/hgn/catkin_ws/src
Build space: /home/hgn/catkin_ws/build
Devel space: /home/hgn/catkin_ws/devel
Install space: /home/hgn/catkin_ws/install
####
#### Running command: "cmake /home/hgn/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/home
/hgn/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/home/hgn/catkin_ws/install -G Unix
Makefiles" in "/home/hgn/catkin_ws/build"
####
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- Check for working C compiler: /usr/bin/cc
```

使工作空间中的环境变量生效

```
source devel/setup.bash
```

3.2 构建baxter环境

首先，为ubuntu安装git

```
sudo apt install git
```

安装成功后，我们可以直接clone baxter官方的包来构建仿真环境

进入到src文件夹里，我们将依次clone，baxter, baxter_common, baxter_tools, baxter_interface, baxter_simulator 这五个项目git clone到ros工作空间的src文件夹下

```
git clone https://github.com/RethinkRobotics/baxter.git
```

```
git clone https://github.com/RethinkRobotics/baxter_common.git
```

```
git clone https://github.com/RethinkRobotics/baxter_tools.git
```

```
git clone https://github.com/RethinkRobotics/baxter_interface.git
```

```
git clone https://github.com/RethinkRobotics/baxter_simulator.git
```

现在src文件夹内部如下图所示：
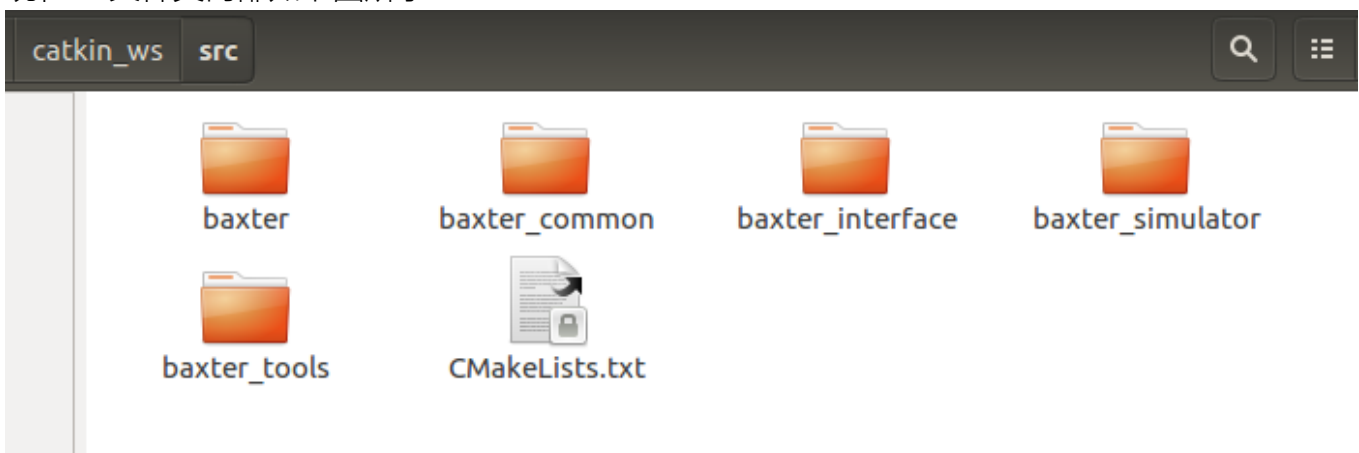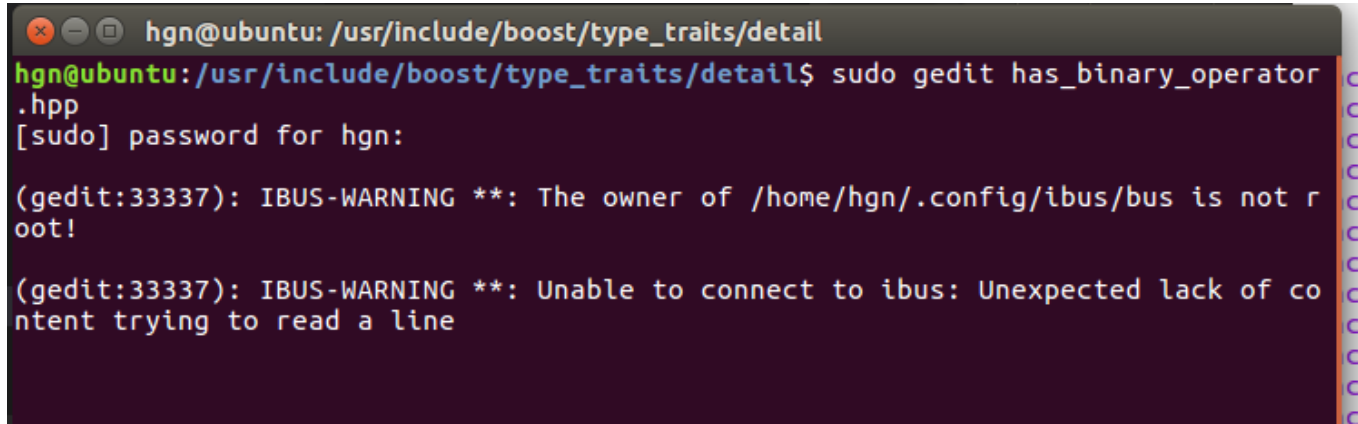
3.3 修复编译错误

接下来我们要修复两个错误，才能正确编译

错误一 <Parse error at "BOOST_JOIN">

首先，我们进入到 has_binary_operator.hpp文件中来修改

```
cd ~
cd /usr/include/boost/type_traits/detail
sudo gedit has_binary_operator.hpp
```


一共要在对应的位置添加画红线的四句话，添加之后保存关闭

错误二 ISO C++ 2011 standard

如下图所示


```
          :o:dir/qfc_sim_io.cxx.o
[ 94%] Building CXX object baxter_simulator/baxter_sim_io/CMakeFiles/baxter_sim_
io.dir/include/baxter_sim_io/moc_qnode.cxx.o
In file included from /usr/include/c++/5/random:35:0,
                 from /usr/include/ignition/math2/ignition/math/Rand.hh:20,
                 from /usr/include/ignition/math2/ignition/math.hh:18,
                 from /usr/include/sdformat-4.0/sdf/Param.hh:34,
                 from /usr/include/sdformat-4.0/sdf/Element.hh:24,
                 from /usr/include/sdformat-4.0/sdf/sdf.hh:5,
                 from /usr/include/gazebo-7/gazebo/common/Battery.hh:25,
                 from /usr/include/gazebo-7/gazebo/common/common.hh:8,
                 from /usr/include/gazebo-7/gazebo/gazebo_core.hh:19,
                 from /usr/include/gazebo-7/gazebo/gazebo.hh:20,
                 from /opt/ros/kinetic/include/gazebo_ros_control/gazebo_ros_con
trol_plugin.h:52,
                 from /home/hgn/catkin_ws/src/baxter_simulator/baxter_gazebo/src
/baxter_gazebo_ros_control_plugin.cpp:42:
/usr/include/c++/5/bits/c++0x_warning.h:32:2: error: #error This file requires c
ompiler and library support for the ISO C++ 2011 standard. This support must be
enabled with the -std=c++11 or -std=gnu++11 compiler options.
 #error This file requires compiler and library support \
  ^
```

打开src文件夹中的CMakeLists.txt, 进行修改
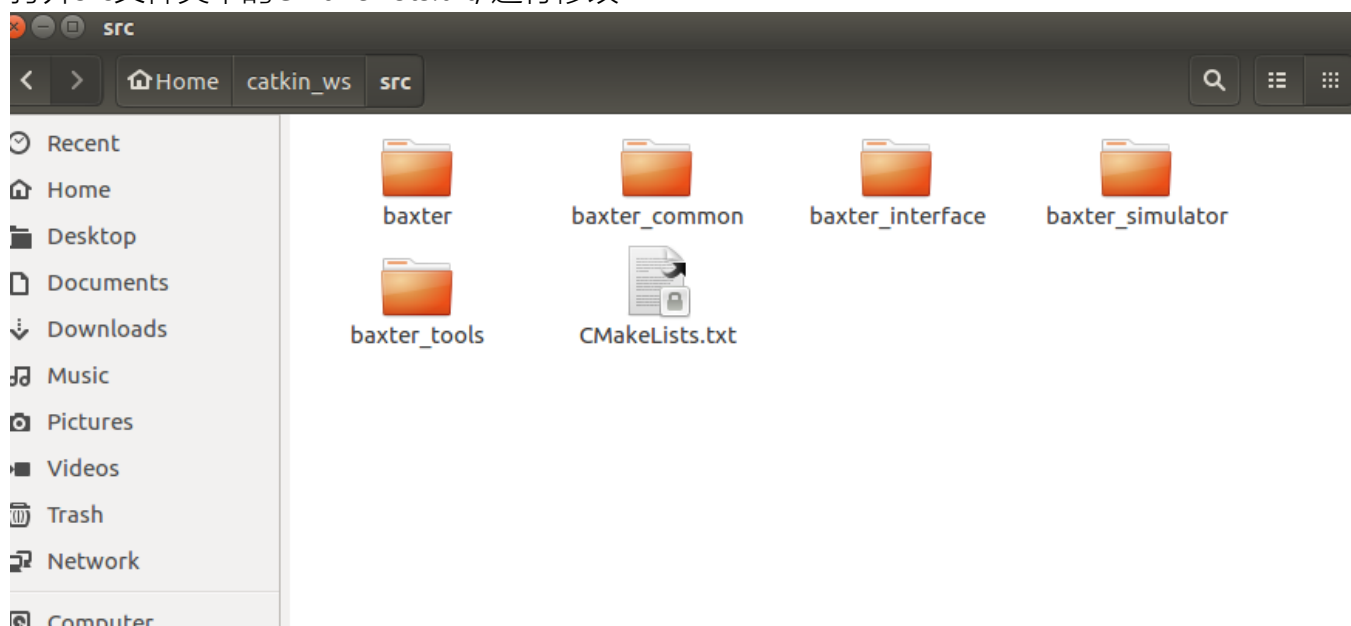

```
sudo gedit CMakeLists.txt
```

在开头加上下面这句，保存并退出

```
SET(CMAKE_CXX_FLAGS "-std=c++0x")
```


开始编译！：

```
cd ~/catkin_ws
catkin_make
```

编译成功！

3.4 运行仿真程序

现在，我们可以用roboware来打开配置好仿真
环境的工作空间！
roboware -> Open Workspace->catkin_ws

在roboware里面打开终端

```
source devel/setup.bash
```

```
roslaunch baxter_sim_examples baxter_pick_and_place_demo.launch
```

哈哈！demo程序成功运行，这个demo是baxter夹取小物块的仿真（从左夹到右，往复运行）

4 下一步是什么？

修改 ik_pick_and_place_demo.py 生成您的自定义baxter仿真程序。

```
     block_poses = list()
270      # The Pose of the block in its initial location.
271      # You may wish to replace these poses with estimates
272      # from a perception node.
273      block_poses.append(Pose(
274          position=Point(x=0.7, y=0.15, z=-0.129),
275          orientation=overhead_orientation))
276      # Feel free to add additional desired poses for the object.
277      # Each additional pose will get its own pick and place.
278      block_poses.append(Pose(
279          position=Point(x=0.75, y=0.0, z=-0.129),
280          orientation=overhead_orientation))
281      # Move to the desired starting angles
282      pnp.move_to_start(starting_joint_angles)
283      idx = 0
284      while not rospy.is_shutdown():
285          print("\nPicking...")
286          pnp.pick(block_poses[idx])
287          print("\nPlacing...")
288          idx = (idx+1) % len(block_poses)
289          pnp.place(block_poses[idx])
290      return 0
291
292  if __name__ == '__main__':
293      sys.exit(main())
294
```

如果发现进行仿真很卡怎么办?

1. **不采用虚拟机的形式来仿真**
   用电脑单独运行ubuntu系统的方式来仿真，这需要您在ubuntu系统上重新安装配置仿真环境，步骤内容与在虚拟机上相同。

2. **我就想要用虚拟机来仿真**
   虚拟机的提速
   我的Gazebo版本7.0

   解决办法：
   以Vmware为例，先关闭虚拟机，打开虚拟机的设置，在显示器栏，勾选加速3D图形，并分配图形内存，我分了8个G
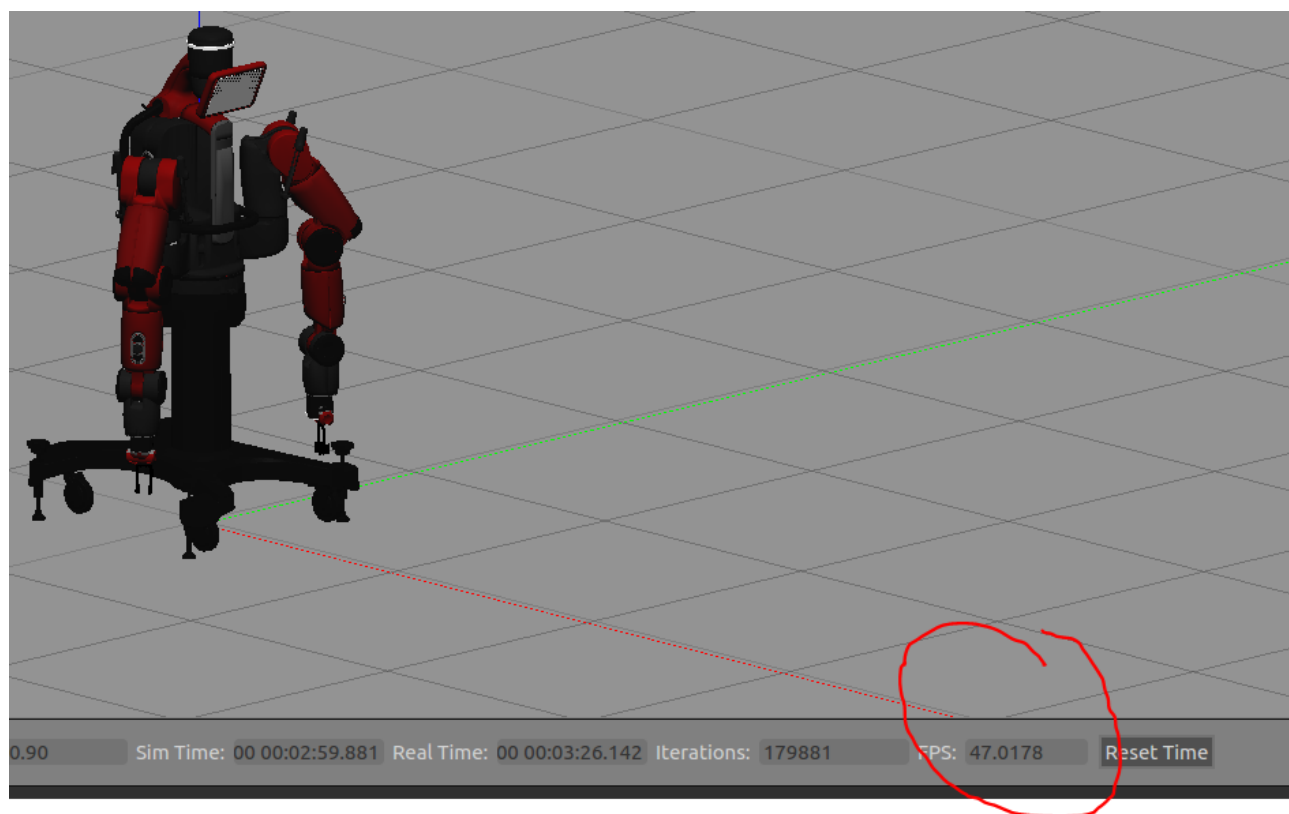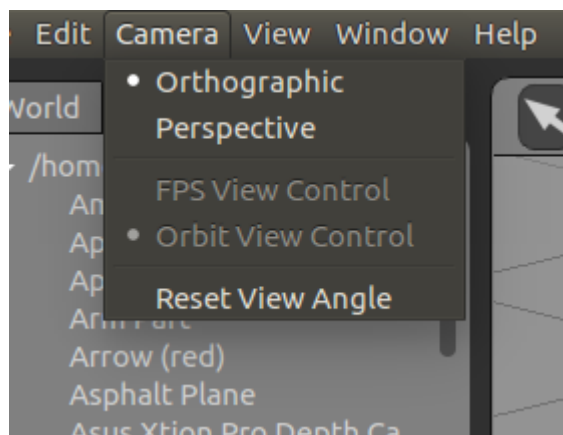
接下来开启虚拟机

此时如果直接运行Gazebo，Gazebo会一直黑屏，还需要设置一下

打开终端：

输入如下代码

```
echo "export SVGA_VGPU10=0" >> ~/.profile
```

```
source ~/.profile
```

相关链接：http://shequ.dimianzhan.com/articles/700/vote_count?

启动Gazebo，此时可以运行到20多帧，接下来在Gazebo左上角Camera选项选择Orthographic，帧率可以进一步提升

最终达到了将近50FPS，已经比较不卡了