《ROS机器人开发技术》

实训讲稿

|  |  |  |
| --- | --- | --- |
| 课程名称 | ： | ROS机器人开发技术 |
| 教师姓名 | ： | 刘梦倩 |
| 提交时间 | ： | 2018年7月15日 |

中国大学MOOC制

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程  名称 | 《ROS机器人开发技术》 | | 章 | | 第三章 | | | 课程类型 |
| 节 | | 第一节 | | | 授课( )  实训(√ ) |
| 名称 | ROS通讯架构（一） | | | | | | |
| 教师 |  | 课时 | | | | 1课时 | | |
| 参考  资料 | 1. ROS wiki <http://wiki.ros.org/> 2. 《ROS机器人编程实践》 3. 《ROS机器人开发实践》 | | | | | | | |
| 教学  目的  要求 | 掌握：ROS的通信架构之node,master,launch | | | | | | | |
| 教学  重点  难点 |  | | | PPT页面 | | | 时间分配 | |
| 教学重点 | | | | | | | |
| 1. 通讯架构的应用 | | | 2~14页 | | | 45分钟 | |
|  | | |  | | |  | |
|  | | |  | | |  | |
| 教学难点 | | | | | | | |
| 1. 通讯架构的应用 | | | 2~14页 | | | 45分钟 | |
|  | | |  | | |  | |
|  |  | | |  | | |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| 教学方法 | | | |
| 本授课以课堂讲授为主，与课堂实时练习方式相结合 | | | |
| 教学内容 | | 操作演示 | 知识点 |
| **PPT第1页:**  下面我们通过下载的ROS-Academy-for-Beginners包进行整个章节的训练和复习。 | |  |  |
| **PPT第2页:**  下面进行任务5：仿真环境下使用topic控制机器人运动，并且观察相机采集到的图像  子任务1.打开仿真环境，启动模拟场景和机器人 | |  |  |
| **PPT第3页:**  每一次我们运行ROS的节点前，都需要把master启动起来，这样才能够让节点启动和注册。  断点1：操作演示**示例代码1**  然后打开仿真环境，启动模拟场景和机器人 | | 操作演示：示例代码1 1.演示目的：roscore基本使用  2. 打开终端。  3.输入：  $roscore    **结论：ROS Master启动。** 示例代码2  1. 演示目的：打开仿真环境，启动模拟场景和机器人 2. 打开新终端。   3.输入命令：  $roslaunch robot\_sim\_demo robot\_spawn.launch    **结论：打开完毕。** |  |
| **PPT第4页:**  下面思考第一个问题：怎样查看当前模拟器中存在的topic？  请大家自行操作一下。  （过段时间）  好了，老师演示一遍，这个命令很简单，就是查看list的命令。  断点1：操作演示**示例代码1**  看到结果是很长的一串代码。这说明当前有很多topic，我们目前并不需要了解每一个topic都是做什么的，后面会慢慢接触。 | | 操作演示：示例代码1 1.演示目的：查看当前模拟器中存在的topic  2. 打开新终端。  3.输入命令：  $ rostopic list    **结论：安装完毕。** |  |
| **PPT第5页:**  那么如何查看全部订阅者？又如何查看全部发布者？  我们使用到的还是list这个命令，只不过要加上参数。  断点1：操作演示**示例代码1**  发布者的类似。  断点1：操作演示**示例代码2**  在这里再提醒各位同学，如果忘记了命令，要善加使用help，查看帮助文件。人的大脑和记忆是有限的，因此我们才需要工具来方便人类对不对。 | | 操作演示：示例代码1  1. 演示目的：查看全部订阅者   2. 打开新终端。  3.输入命令：  $ rostopic list -s    **结论：查看完毕。** 示例代码2  1. 演示目的：查看全部发布者   2. 打开新终端。  3.输入命令：  $ rostopic list -p    **结论：查看完毕。** |  |
| **PPT第6页:**  我们接下来要使用/camera/rgb/image\_raw这个topic进行通讯练习，大家在自己的list当中找到这个topic，查看他的信息。要如何操作？  断点1：操作演示**示例代码1**  可以看到：这个topic 的type是sensor\_msgs/Image  有一个发布者就是gazabo，但是目前没有订阅者订阅这个话题。 | | 操作演示：示例代码1 1.演示目的：查看topic /camera/rgb/image\_raw  2. 打开终端。  3.输入命令：  $ rostopic info /camera/rgb/image\_raw    **结论：查看完毕。** |  |
| **PPT第7页:**  如果我们只需要查看/camera/rgb/image\_raw这个话题的类型信息，该如何操作。  断点1：操作演示**示例代码1**  可以看出，只显示了类型信息。  下面一个问题：  如何寻找所有sensor\_msgs/Image类型的topic？  断点1：操作演示**示例代码2**  可以看到一共有两个topic是这个类型的。  其中第一个是彩色摄像机的图像信息。  第二个是深度相机的图像信息。 | | 操作演示：示例代码1 1.演示目的：查看topic /camera/rgb/image\_raw的类型  2.输入命令：  $ rostopic type /camera/rgb/image\_raw    **结论：查看完毕。** 示例代码2 1.演示目的：寻找所有sensor\_msgs/Image类型的topic  2.输入命令：  $ rostopic find sensor\_msgs/Image    **结论：查看完毕。** |  |
| **PPT第8页:**  下面看第四个子任务：怎样查看sensor\_msgs/Image类型的信息？。  大家先思考一下自己尝试操作。  下面老师操作一下：  断点1：操作演示**示例代码1**  这个msg包含了很多的信息。包括很多的int、string等等。 | | 示例代码1 1.演示目的：查看/sensor\_msgs/Image类型的信息  2.输入命令：  $ rosmsg show sensor\_msgs/Image    **结论：查看完毕。** |  |
| **PPT第9页:**  复习告一段落了，下面我们做一些新的东西，指定image\_view接收这个消息。  我们可以尝试一下输入：  $ rosrun image\_view image\_view  发现不行，显示这样的信息：    因此我们按照它的指引改变我们的命令：  $ rosrun image\_view image\_view image:=<image topic> [transport]  先Ctrl c退出当前的进程，这样操作：  断点1：操作演示1  最后面transport的部分可以忽略，只改掉尖括号里面的内容。  可以看到弹出一个新的窗口，上面显示的是模拟器中机器人的摄像头拍摄到的场景。我们可以通过前面章节中用到的控制机器人行动的节点来看看显示的景色的变化。 | | 示例代码1 1.演示目的：指定image\_view接收这个消息  2.输入命令：  $ rosrun image\_view image\_view image:=/camera/rgb/image\_raw    **结论：接收完毕。** |  |
| **PPT第10页:**  可以看到弹出一个新的窗口，上面显示的是模拟器中机器人的摄像头拍摄到的场景。我们可以通过前面章节中用到的控制机器人行动的节点来看看显示的景色的变化。 | |  |  |
| **PPT第11页:**  下面看第6个任务：如何控制机器人的运动？。  这部分我们在前面的章节已经做过一次了，我们现在来复习一下：  断点：操作演示1  下面我们就可以通过键盘来控制机器人的移动了，同时我们还可以看到图像信息的部分会随着小车的移动而改变。  这说明我们和机器人之间的通讯良好。 | | 示例代码1  1. 演示目的：控制机器人的运动 2. 打开新终端   3.输入命令：  $ rosrun robot\_sim\_demo robot\_keyboard\_teleop.py    **结论：可以开始控制。** |  |
| **PPT第12页:**  这时候再看看/camera/rgb/image\_raw这个topic的信息，看到有什么区别？  操作还是很简单，刚刚已经做过一遍了，我们在做一遍。  断点：操作演示1  看到多了一个订阅者。  就是我们屏幕上的这个窗口节点。 | | 示例代码1 1.演示目的：/camera/rgb/image\_raw信息  2.打开新终端  3.输入命令：  $ rostopic info /camera/rgb/image\_raw    **结论：可以开始控制。** |  |
| **PPT第13页:**  /camera/depth/image\_raw是深度相机的发布的图像，如何显示这个深度相机的图像呢？  首先我们关闭之前的图像窗口，同学们刚刚已经尝试过了显示彩色相机的图像，请大家自己做一下，如何显示深度相机的？  （过一段时间）  那么大家估计会做的已经做出来了，不会做的跟着老师再做一遍。  断点：操作演示1  得到深度相机的图像后我们再次操纵机器人移动，能大概看出相机对应的场景里面的内容。 | | 示例代码1 1.演示目的：显示深度相机的图像  2.打开新终端  3.输入命令：  $ rosrun image\_view image\_view image:=/camera/depth/image\_raw    **结论：显示深度相机的图像。** |  |
| **PPT第14页:**  问题7：在机器人运动的过程中，如何查看速度指令topic的内容：/cmd\_vel？  断点：操作演示1  操作机器人移动，我们看到终端上不断有信息刷新，这些就是机器人的速度指令topic的信息。这些信息经由我们的键盘传递到模拟器的机器人当中，用来控制机器人的移动。 | | 示例代码1 1.演示目的：查看速度指令  2.打开新终端  3.输入命令：  $ rostopic echo /cmd\_vel  **结论：查看速度指令。** |  |
| PPT第15页: 我们本节课的内容就是这些。希望大家能够在课上复习到我们之前学过的知识，打下良好的基础。 | |  |  |
| 知识点框图 | | | |
| 结构图 |  | | |