7. MoveIt scene design

This lesson takes MoveIT simulation as an example. If you need to set up the real machine and simulation to be synchronized, please see the lesson [02, MoveIt Precautions and Controlling the Real Machine]. ! ! ! be safe! ! !

The effect demonstration is a virtual machine and other main control running conditions (related to the main control performance, depending on the actual situation).

7.1. Start

```
#Raspberry Pi 5 master needs to enter docker first, please perform this step
#If running the script into docker fails, please refer to ROS/07, Docker tutorial
~/run_docker.sh
```

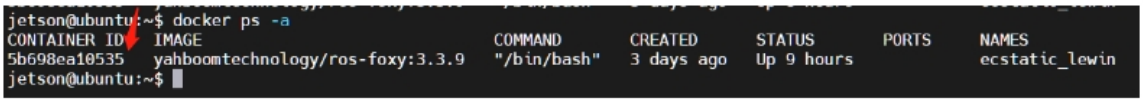
Start MoveIT

```
roslaunch arm_moveit_demo x3plus_moveit_demo.launch sim:=true
```

<PI5 needs to open another terminal to enter the same docker container

1. In the above steps, a docker container has been opened. You can open another terminal on the host (car) to view:

```
docker ps -a
```



2. Now enter the docker container in the newly opened terminal:

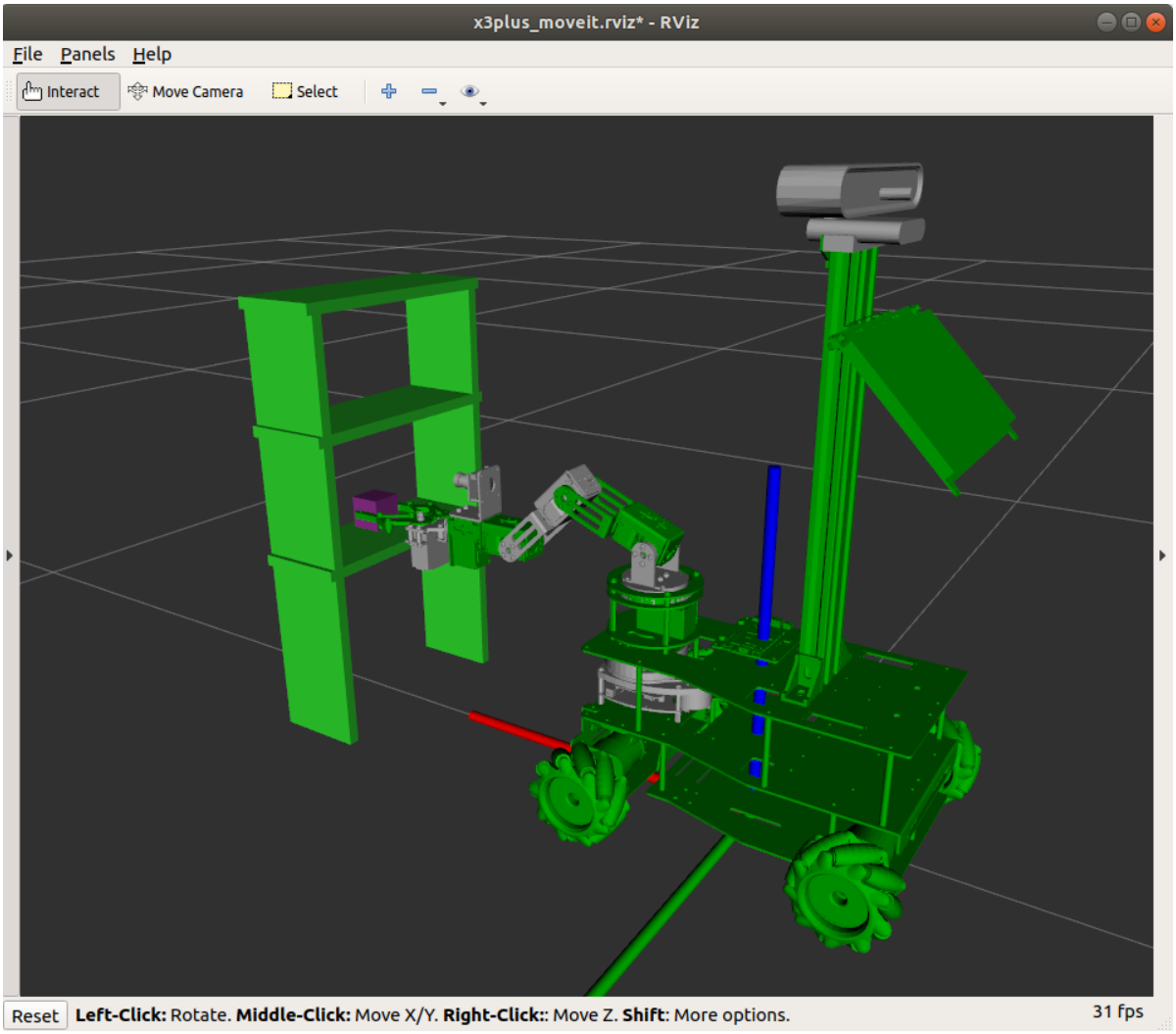
```
docker exec -it 5b698ea10535 /bin/bash
```

After successfully entering the container, you can open countless terminals to enter the container.

Start the scene design node

```
rosrun arm_moveit_demo 06_set_Scene.py
```

The renderings are as follows



7.2, source code

Open the jaws

```
    # Simulation
    pub_joint = rospy.Publisher("/move_group/fake_controller_joint_states",
JointState, queue_size=1000)
    #realmachine
    pub_Arm = rospy.Publisher("TargetAngle", ArmJoint, queue_size=1000)
    arm_joint = ArmJoint()
    arm_joint.id = 6
    arm_joint.angle = 180 - 0.55 * 180 / pi
    joint_state = JointState()
    joint_state.name = ["grip_joint"]
    joint_state.position = [-0.58]
    for i in range(10):
        pub_joint.publish(joint_state)
        pub_Arm.publish(arm_joint)
        sleep(0.1)
```

Add the robot arm end clamp block

```
    p = PoseStamped()
    p.header.frame_id = end_effector_link
    p.pose.orientation.w = 1
    # Add tool
    scene.attach_box(end_effector_link, 'tool', p, [0.03, 0.03, 0.03])
```

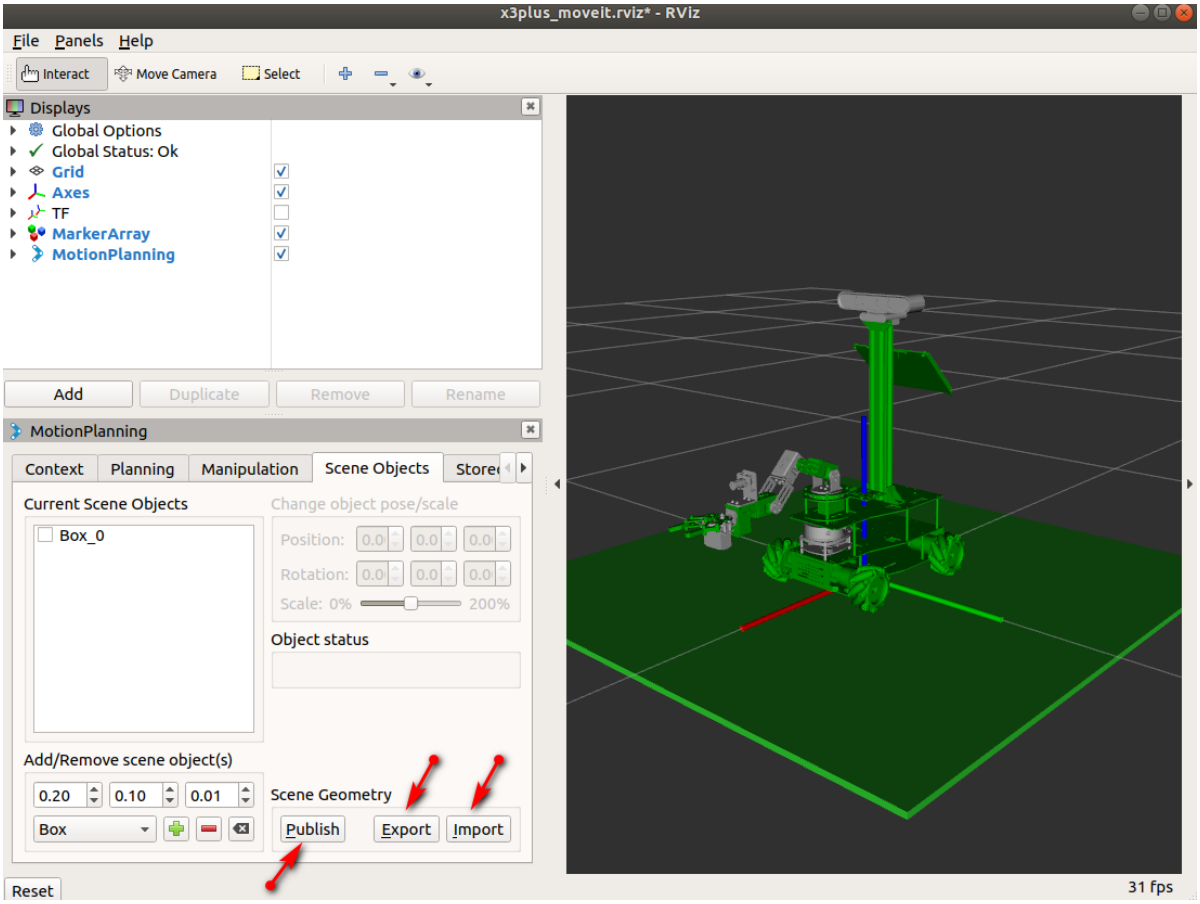
Add supports

```
    table_list = {
        "obj0": [[0.08, 0.01, 0.4], [0.4, -0.1, 0.2]],
        "obj1": [[0.08, 0.01, 0.4], [0.4, 0.1, 0.2]],
        "obj2": [[0.08, 0.22, 0.01], [0.4, 0, 0.4]],
        "obj3": [[0.08, 0.22, 0.01], [0.4, 0, 0.29]],
        "obj4": [[0.08, 0.22, 0.01], [0.4, 0, 0.17]],
    }
    # add obj
    for i in range(len(table_list)):
        add_obj(p, table_list.keys()[i], table_list[table_list.keys()[i]][0],
                table_list[table_list.keys()[i]][1])
```

Loop planning between two points

```
    i = 0
    while i < 5:
        yahboomcar.set_joint_value_target(target_joints1)
        yahboomcar.go()
        yahboomcar.set_joint_value_target(target_joints2)
        yahboomcar.go()
        i += 1
        print ("{}th planning!!!".format(i))
```

7.3. Scene import and export



7.3.1. Import

As shown above, click the [Import] button, select the [arm_moveit_demo/scene/floor.scene] file, and confirm it to import.

Be sure to click [Publish], otherwise it will not work.

7.3.2. Export

As shown above, click the [Export] button, select the path to be saved, and modify the name to be saved. It will be used for import and release next time.