6.Collision detection

1. Robotic arm rviz simulation motion

- **Start virtual machine emulation**

Start MoveIT (virtual machine side)

```
roslaunch dofbot_config demo.launch
```

Start collision detection node

```
rosrun dofbot_moveit 03_attached_object.py     # python file
```

Close case: [ctrl+c] to close. If it cannot be closed, execute [ctrl+z] again.

Method to create obstacles: Click SceneObjects in MotionPlanning to enter the scene settings. We select the Box attribute, adjust the scene object(s) parameters, click the plus sign to add obstacles; click the minus sign to delete obstacles. Move the obstacle to the range of your robotic arm, as shown below.
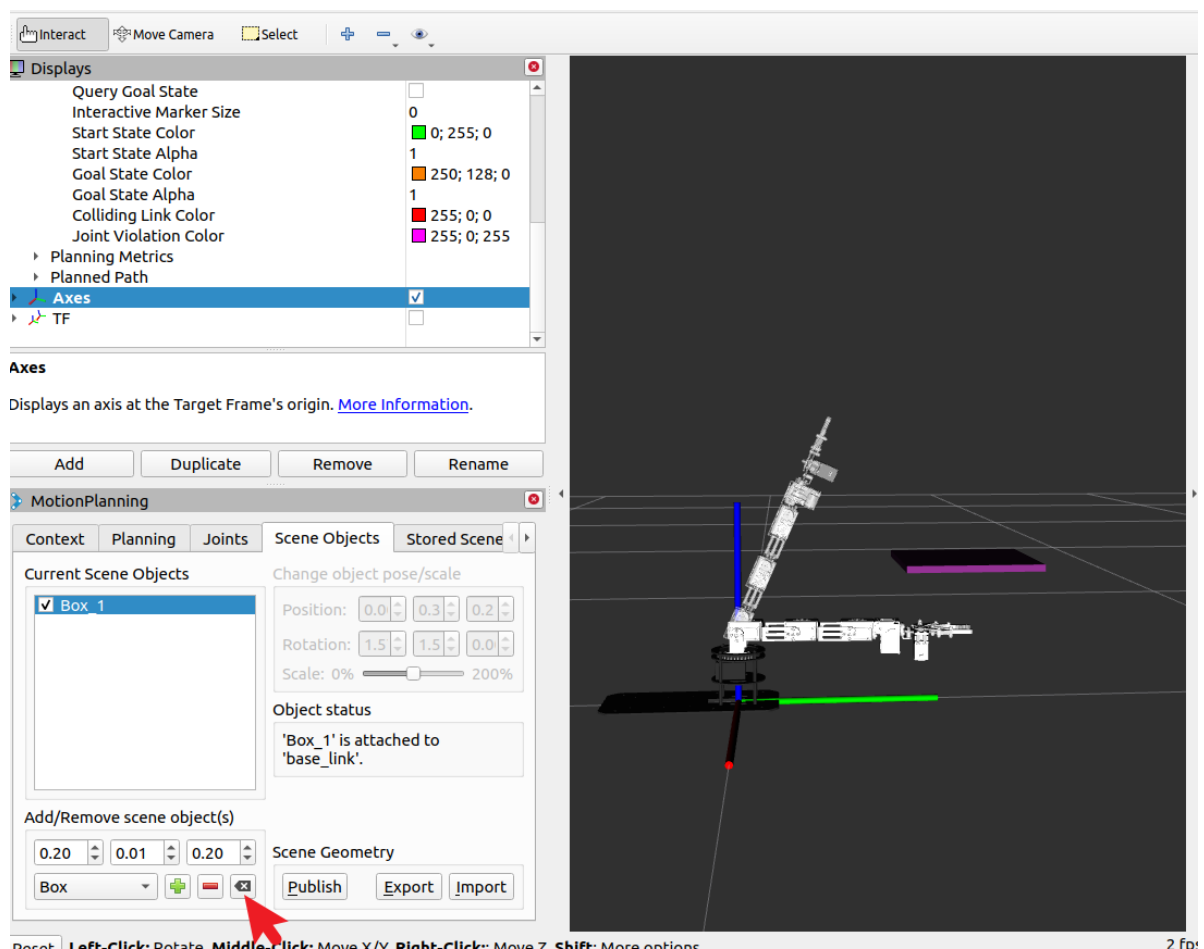
You can also adjust the Position and Rotation attributes, change the position of the box, add two boxes Box_1, click [Publish] after the addition is completed to take effect; click the [Export] button to export the scene and save it to dofbot_ws/src/dofbot_config/scene /floor.scene; the next time you use it, just click [Import] to import.

Code design:

Add an obstacle with a length, width and height of [0.7, 0.1, 0.02] in front of the robotic arm. The center point of the obstacle is located at xyz(0,0.15,0.21)

```
# Initialize the scene object
scene = PlanningSceneInterface()
# Set the height of the desktop
table_ground = 0.2
# Set the 3D dimensions of the obstacle [length, width, and height]
table_size = [0.7, 0.1, 0.02]
# Add the table to the scene
table_pose = PoseStamped()
table_pose.header.frame_id = 'base_link'
table_pose.pose.position.x = 0
table_pose.pose.position.y = 0.15
table_pose.pose.position.z = table_ground + table_size[2] / 2.0
table_pose.pose.orientation.w = 1.0
```

Code path: dofbot_ws/src/dofbot_moveit/scripts/03_attached_object.py

Experimental phenomenon: You can see that the robotic arm will avoid obstacles in rviz.
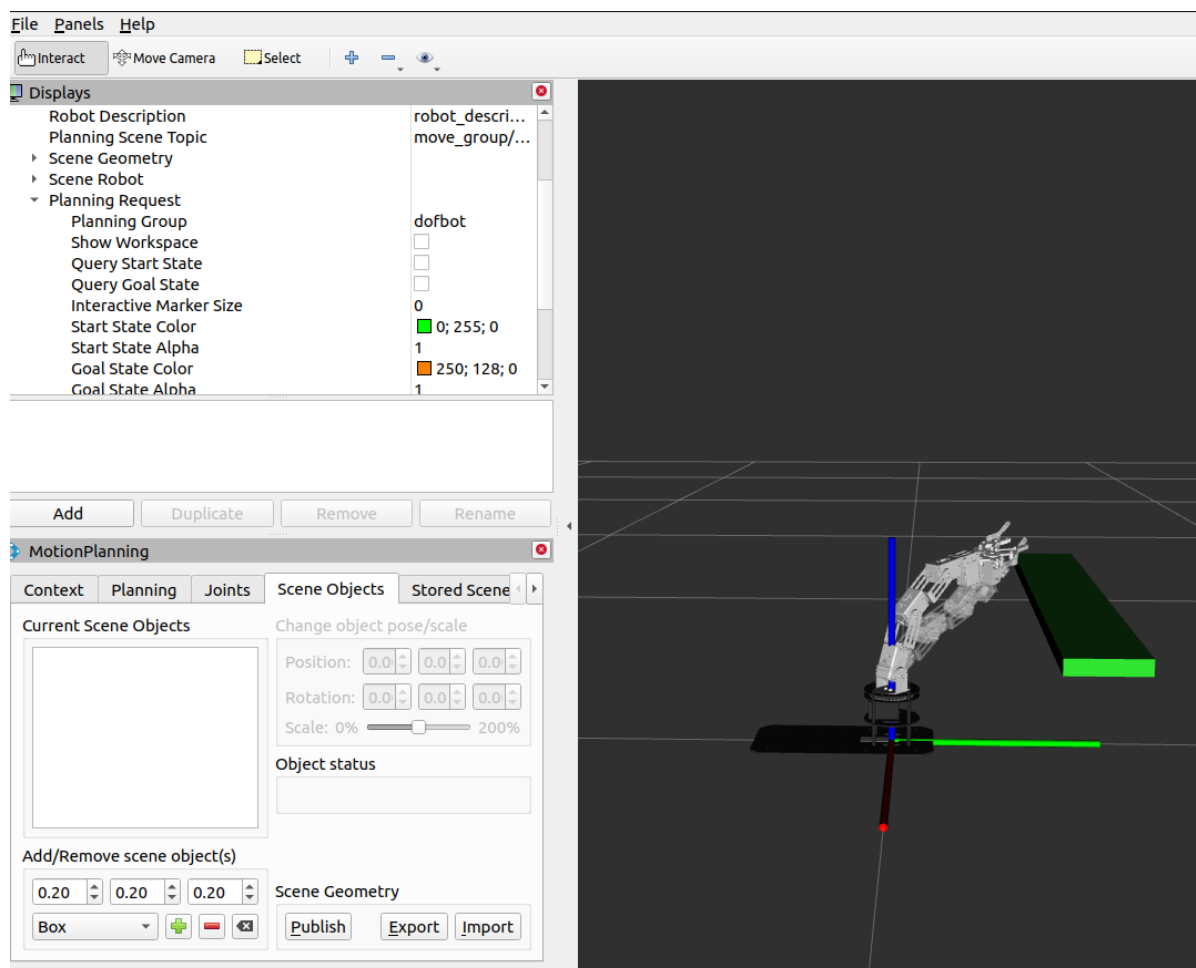
2.Robotic arm rviz simulation motion+real robotic arm operation

Close the previous process, and open another terminal and enter the following command (the effect of this program is to simulate real machine motion in rviz)

```
rosrun dofbot_moveit 3.py # python file
```

Close case: 【 ctrl+c 】 Close, cannot close, then execute 【 ctrl+z 】.

Key part program code description:

```
#Subscribe to the angles of each joint published by Movelt
subscriber = rospy.Subscriber("/joint_states", JointState, topic)
#The subscribed angle is calculated and transmitted to the Python driver library
for i in range(6):
        joints[i] = (msg.position[i] * RA2DE) + 90
        if(i == 5):
            joints[i] = (msg.position[i] * 116) + 180
#Call driver function
sbus.Arm_serial_servo_write6_array(joints, 100)
```

Code path: dofbot_ws/src/dofbot_moveit/scripts/3.py

Experimental phenomenon: The real machine robotic arm will move along with the model in rviz.