6. MoveIt Collision detection

This lesson takes the MoveIT simulation as an example. If you need to set the synchronization between the real machine and the simulation, please refer to the lesson [02, MoveIt Precautions and Controlling the Real Machine]. ! ! ! be careful! ! !

The effect demonstration is a virtual machine, and other masters are running (related to the performance of the master, depending on the actual situation).

6.1. Start

Start the MoveIT

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

Start collision detection node

```
rosrun arm_moveit_demo 05_attached_object    # C++
```

Set up random movements for about 10 times. Since the target position is random, it may also be randomly placed on the obstacle. When the target position comes into contact with the obstacle, the planning will fail.

The effect diagram is as follows

6.2. Source code

6.2.1. C++ files

Set the size and position of obstacles

```
    moveit::planning_interface::PlanningSceneInterface scene;
    ////////////////////////Add obstacles///////////////////////
    vector<string> object_ids;
    scene.removeCollisionObjects(object_ids);
    vector<moveit_msgs::CollisionObject> objects;
    moveit_msgs::CollisionObject obj;
    // Set the ID of the obstacle
    obj.id = "obj";
    object_ids.push_back(obj.id);
    // The status of the obstacle
```

```
    obj.operation = obj.ADD;
    obj.header.frame_id = frame;
    shape_msgs::SolidPrimitive primitive;
    // Set obstacle type
    primitive.type = primitive.BOX;
    // Set obstacle dimensions
    primitive.dimensions.resize(3);
    //Set the length, width and height of the obstacle
    primitive.dimensions[0] = 0.2;
    primitive.dimensions[1] = 0.1;
    primitive.dimensions[2] = 0.02;
    obj.primitives.push_back(primitive);
    geometry_msgs::Pose pose;
    // Set the location information of obstacles[x,y,z]
    pose.position.x = 0.2;
    pose.position.y = 0;
    pose.position.z = 0.26;
    tf::Quaternion quaternion;
    // The units of R, P and Y are angles
    double Roll = 0.0;
    double Pitch = 0.0;
    double Yaw = 90.0;
    quaternion.setRPY(Roll * M_PI / 180, Pitch * M_PI / 180, Yaw * M_PI / 180);
    pose.orientation.x = quaternion.x();
    pose.orientation.y = quaternion.y();
    pose.orientation.z = quaternion.z();
    pose.orientation.w = quaternion.w();
    // Set the pose information of obstacles
    obj.primitive_poses.push_back(pose);
    objects.push_back(obj);
```

Set the color of the obstacle

```
    ////////////////////////Set the color of the
obstacle////////////////////////
    // Create a color container for detection objects
    std::vector<moveit_msgs::ObjectColor> colors;
    moveit_msgs::ObjectColor color;
    // Add the id whose color needs to be set
    color.id = "obj";
    //Set RGBA value, range [0~1]
    color.color.r = 1.0;
    color.color.g = 0;
    color.color.b = 0;
    color.color.a = 0.5;
    colors.push_back(color);
    // Add set information to the scene
    scene.applyCollisionObjects(objects, colors);
```

Set random movement

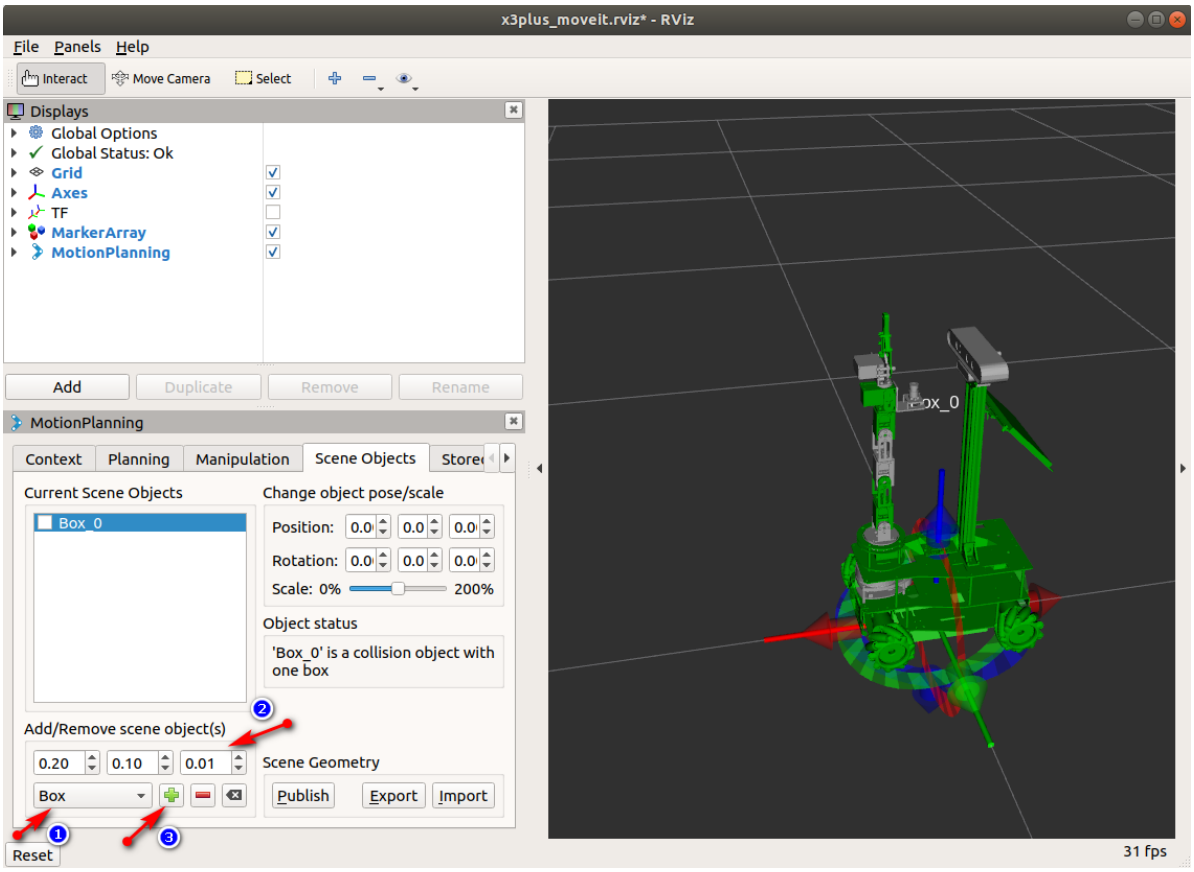
```
    int index = 0;
    while (index < 10) {
        yahboomcar.setRandomTarget();
        yahboomcar.move();
        sleep(0.5);
        index++;
        cout << "The " << index << " plan!!!" << endl;
    }
```

6.3、Scene Object

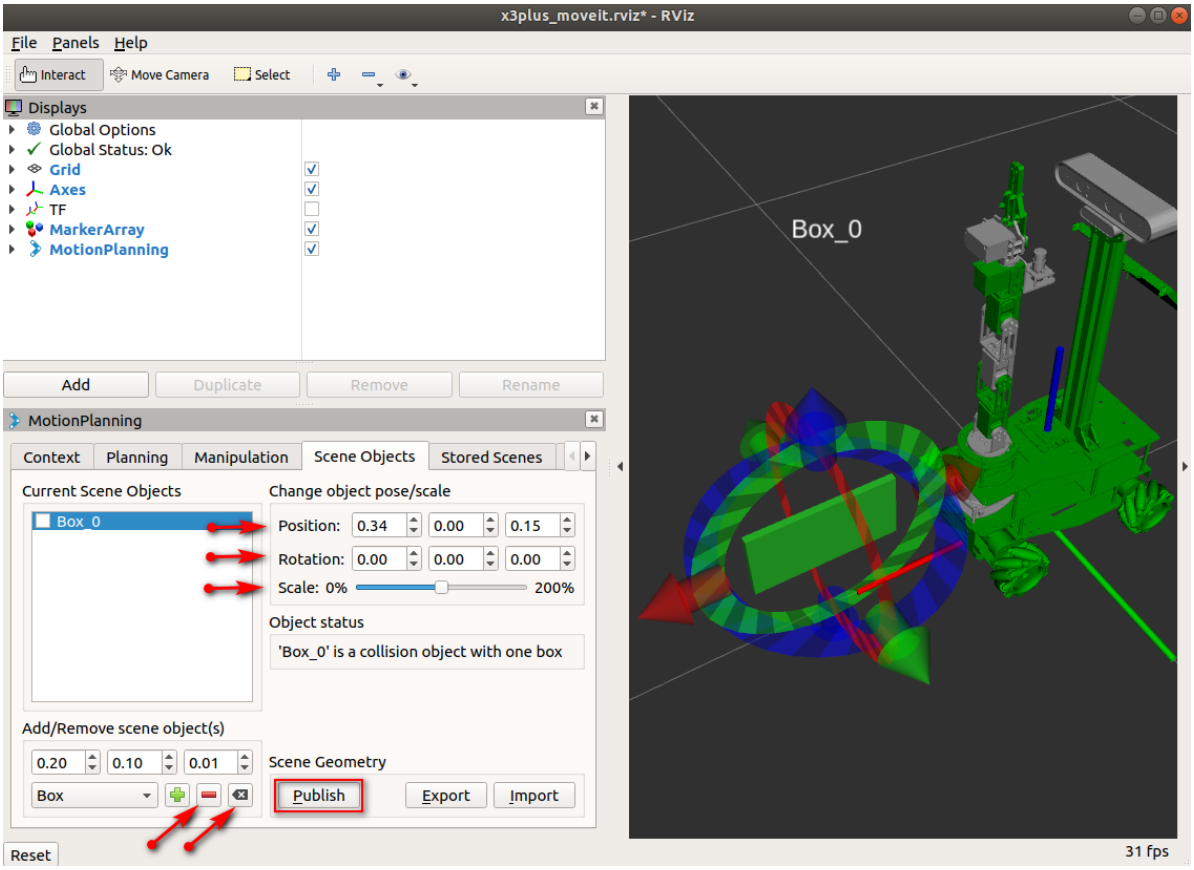
In addition to using code to add or delete obstacles, you can also use the [MotionPlanning] tool in rviz to add or delete.

6.3.1. Add obstacles

The first step is to select the shape of the obstacle to be added [for example: Box], then set the size of the obstacle, and then click the [+] sign. At this time, the position of the obstacle is at the center of the origin. As shown below



In the second step, drag the arrow on the right to drag out the obstacle. You can also directly set the [Position] position information, [Rotation] posture information, [Scale] scaling ratio, etc. As shown below



Step 3: After completing the above setting steps, be sure to click [Publish] for it to take effect; otherwise, the robot may pass through obstacles.

6.3.2. Delete obstacles

As shown above,

Delete: Select the obstacle first, then click the [-] sign to delete the current obstacle.

Clear: Click the [x] sign directly to clear the obstacles.