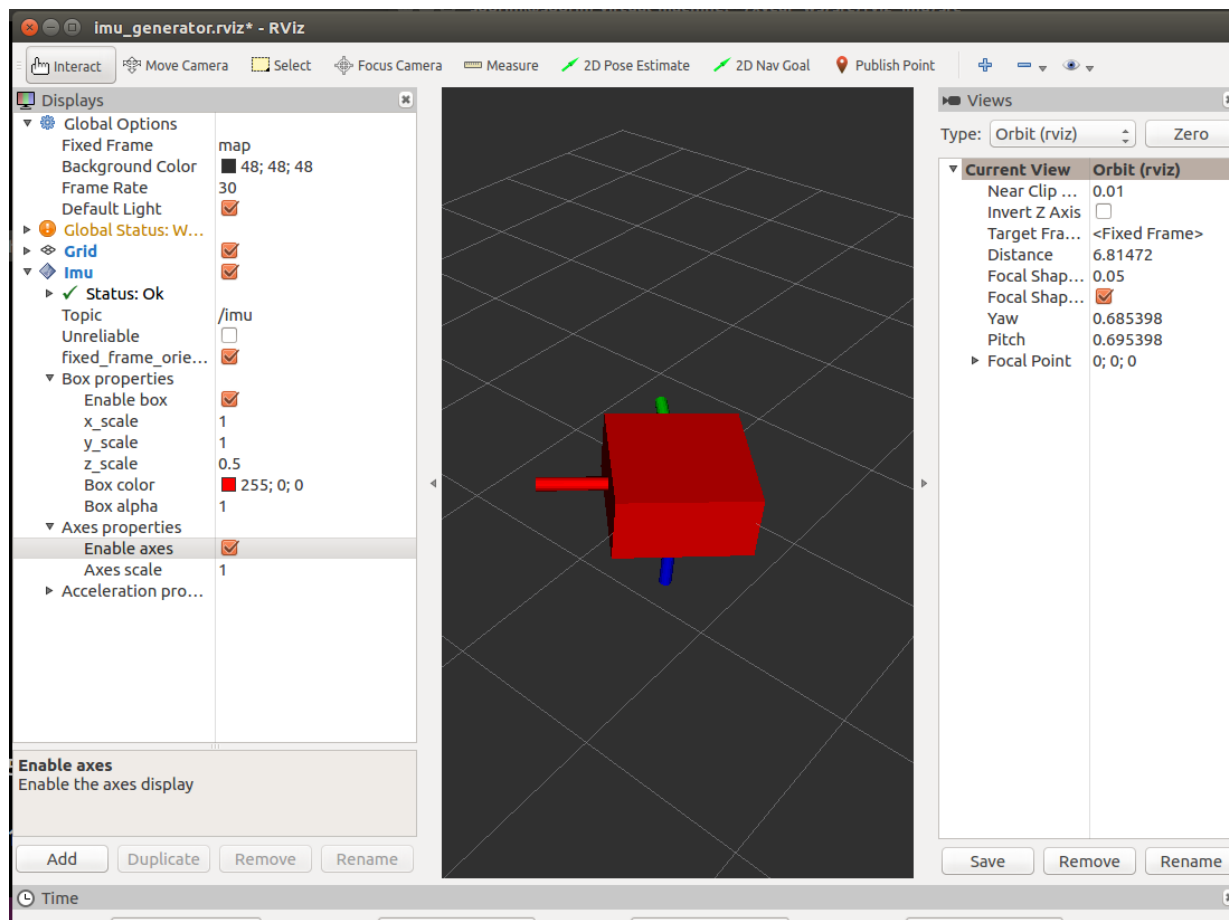


RVIZ기반 IMU뷰어 제작

과제 설명

- IMU데이터로 뷰어의 박스를 움직이기
 - Imu_data.txt -> imu_generator.py -> RVIZ뷰어

현상 확인 #1



현상 확인 #2

```
header:
  seq: 181
  stamp:
    secs: 1610468216
    nsecs: 233643054
  frame_id: "map"
orientation:
  x: 0.967664031427
  y: -0.188173417712
  z: 0.164613713547
  w: 0.0334576218798
orientation_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
angular_velocity:
  x: 0.0
  y: 0.0
  z: 0.0
angular_velocity_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
linear_acceleration:
  x: 0.0
  y: 0.0
  z: 0.0
linear_acceleration_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
---
```

코드 분석 #1

- `rospy.init_node('imu_generator')`
 - `Imu_generator`라는 이름으로 노드 설정
- `pub=rospy.Publisher('imu', Imu, queue_size=10)`
 - `Imu type`의 메시지를 발행하도록 설정
- `rate=rospy.Rate(10)`
 - 초마다 10번의 토픽을 전송하도록 설정

코드 분석 #2

- `f=open("/home/soorim/xycar_ws/src/rviz_imu/src/imu_data.txt", 'r')`
- `lines=f.readlines()`
 - `Imu_data.txt`를 열고 한 줄 씩 읽음
- `imuMsg = Imu()`
 - `imuMsg`를 `imu type`으로 설정
- `imuMsg.header.frame_id = 'map'`
- `seq=0`

코드 분석 #3

```
for line in lines:
```

```
    item=line.split(",")
```

```
    roll=float(item[0][7:21])
```

```
    pitch=float(item[1][9:23])
```

```
    yaw=float(item[2][7:21])
```

- 해당 파일을 한 줄 씩 읽고 콤마를 기준으로 나눈 후 roll, pitch, yaw 값 지정

코드 분석 #4

- `quat = quaternion_from_euler(roll,pitch,yaw)`
 - Roll, pitch, yaw 값을 quaternion값으로 변환
- `imuMsg.orientation.x = quat[0]`
- `imuMsg.orientation.y = quat[1]`
- `imuMsg.orientation.z = quat[2]`
- `imuMsg.orientation.w = quat[3]`
 - 해당 값을 imuMsg에 x,y,z,w에 맞게 지정
- `pub.publish(imuMsg)`
 - imuMsg 메시지 발행