**Robot Autonomy Homework 0**

Name: Naman Kumar  
Andrew ID: namank

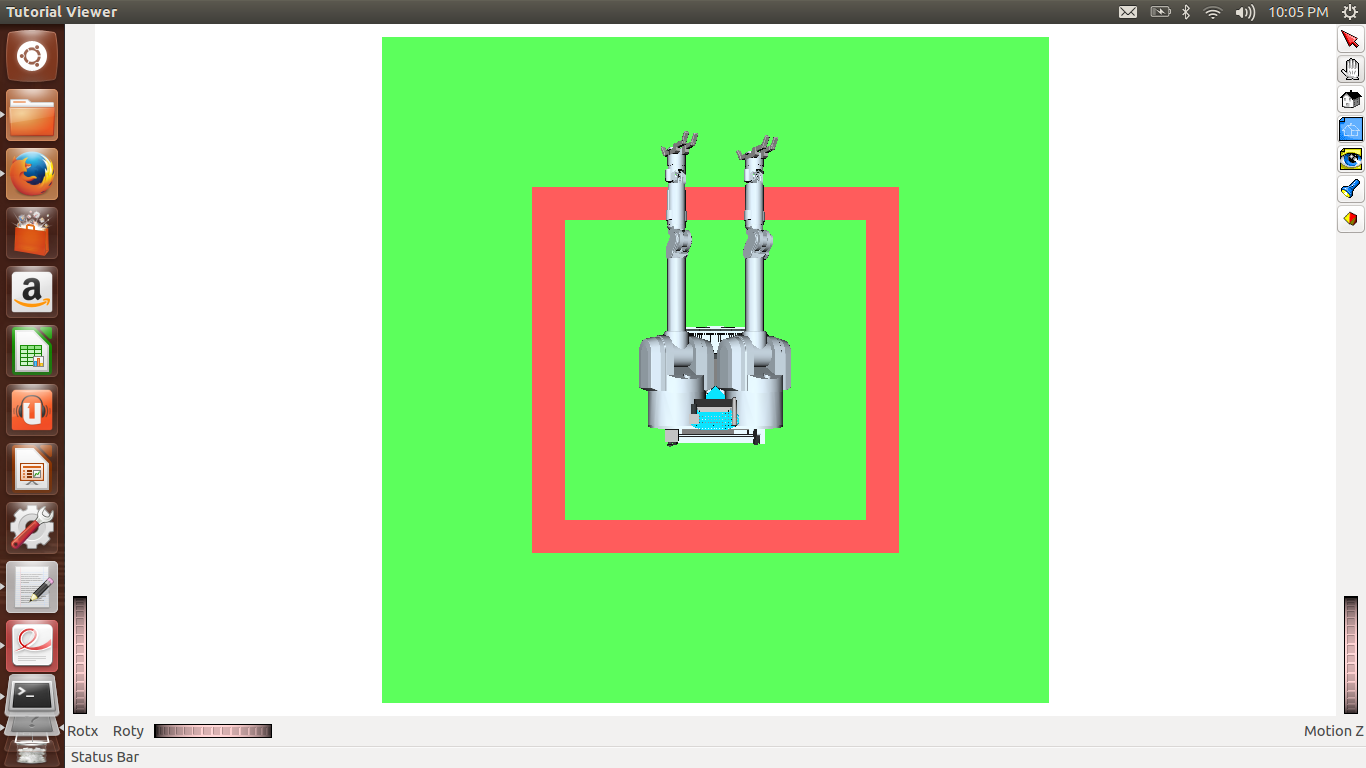
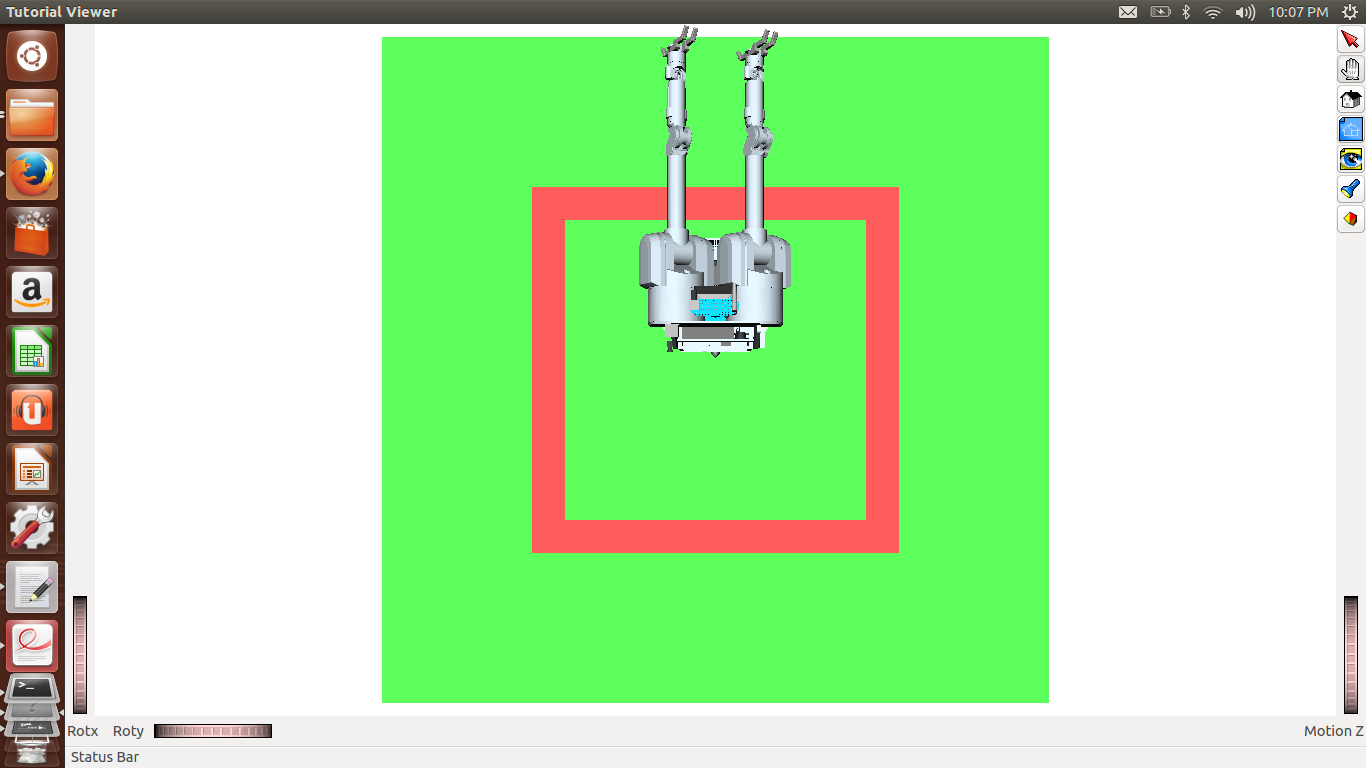
Python Sorting Functions  
All the three sorting functions are in the file **python\_tutorial.py** and the output is printed in the format mentioned in the Homework Problem set.

OpenRave Functions  
**1. move\_straight**  
a) Code:  
***with self.env:***

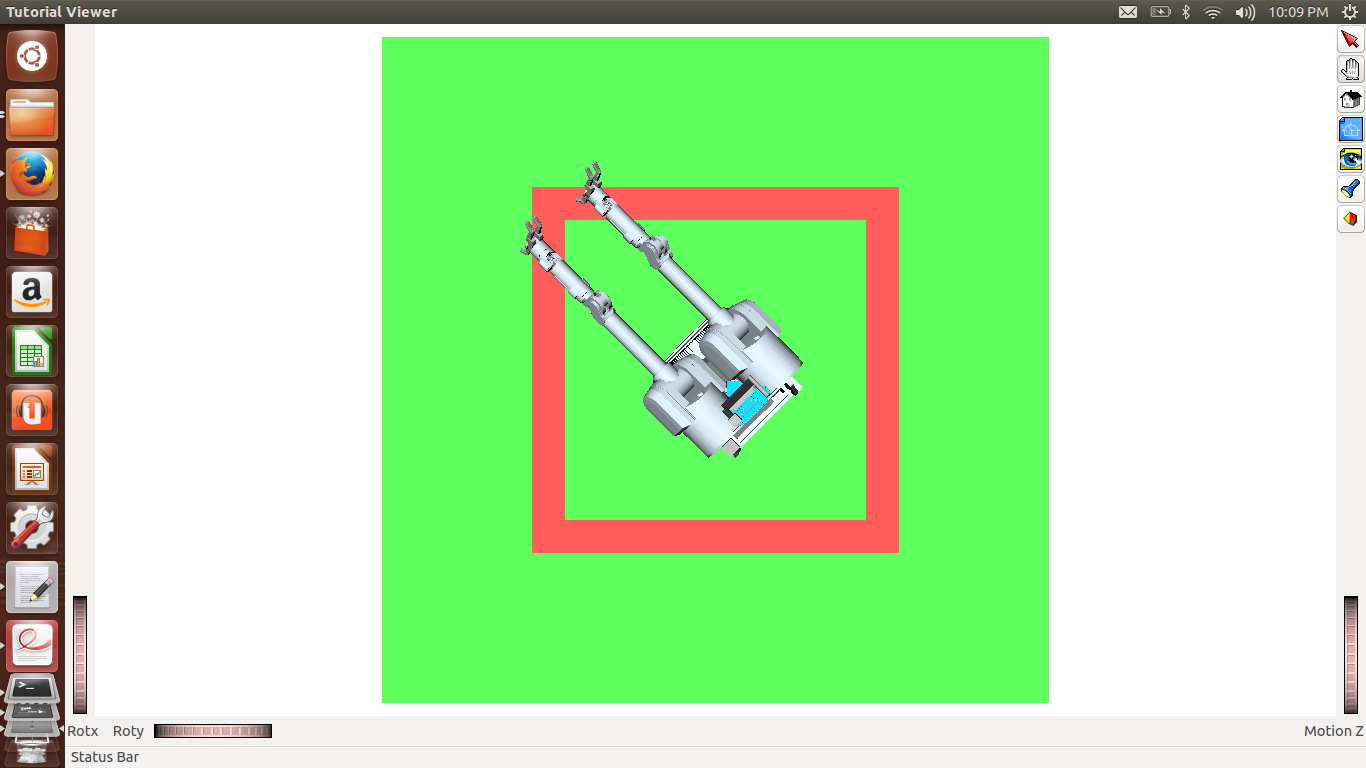
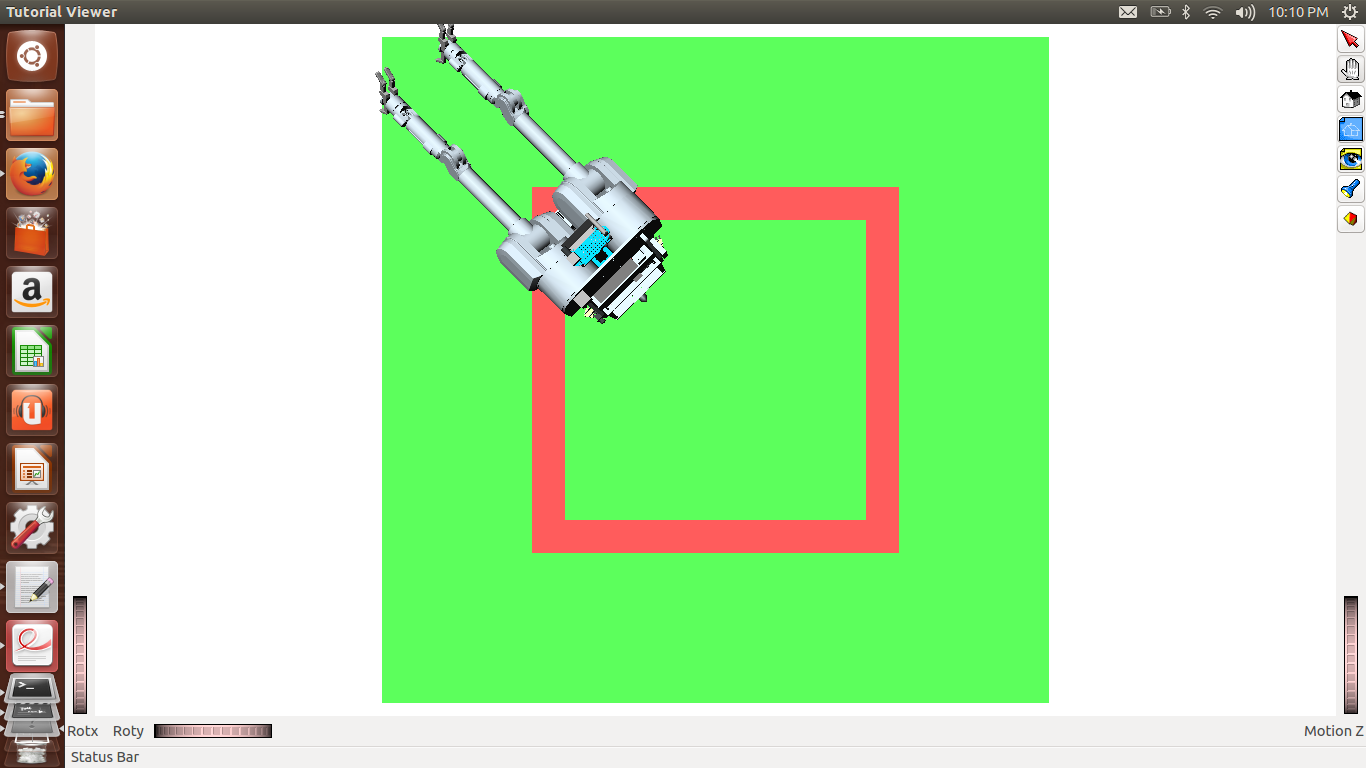
***T1 = self.robot.GetTransform() # Get the current Transform  
T2 = openravepy.axisAngleFromRotationMatrix(T1[0:3,0:3]) # Get current axis angle  
T1[0,3] = dist\*np.cos(T2[2]) # Distance moved in X-direction  
T1[1,3] = dist\*np.sin(T2[2]) # Distance moved in Y-direction  
self.robot.SetTransform(T1) # Setting the new transform***

b) Images:

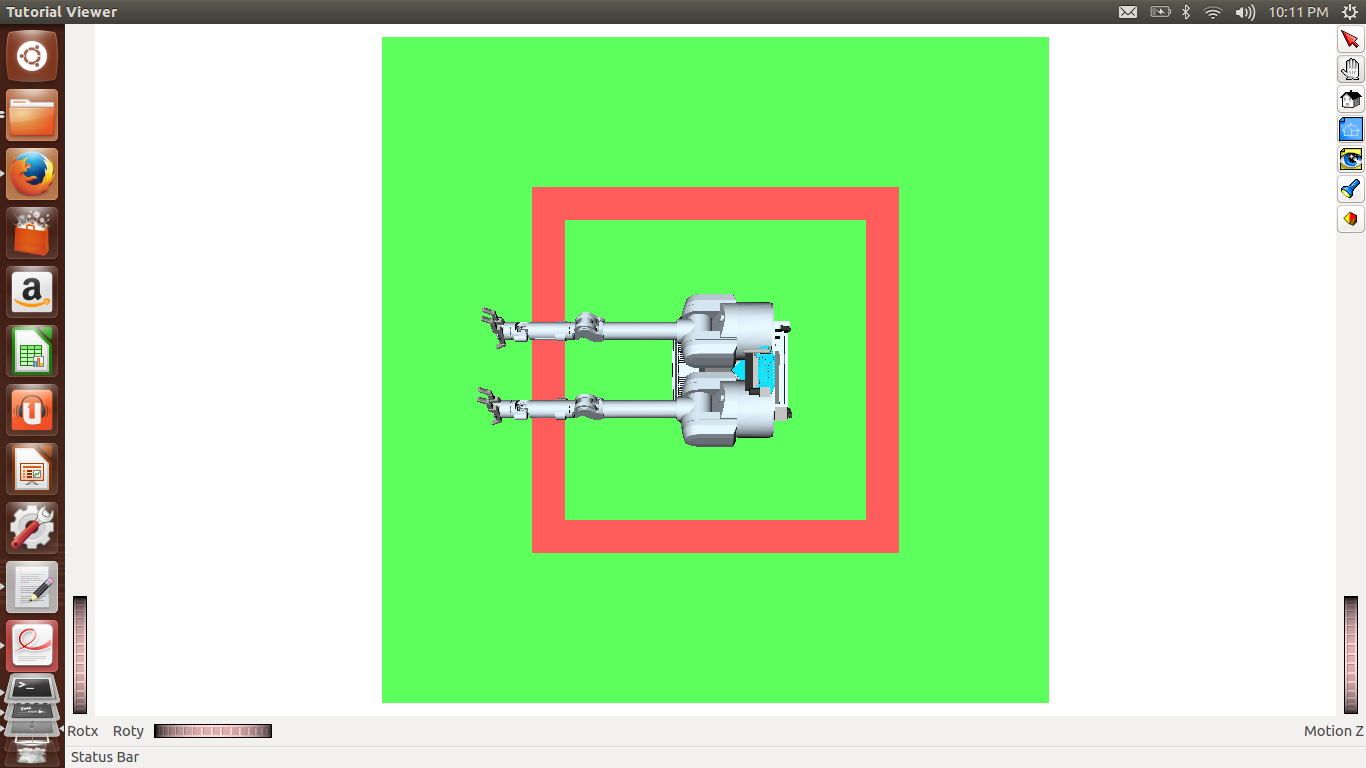
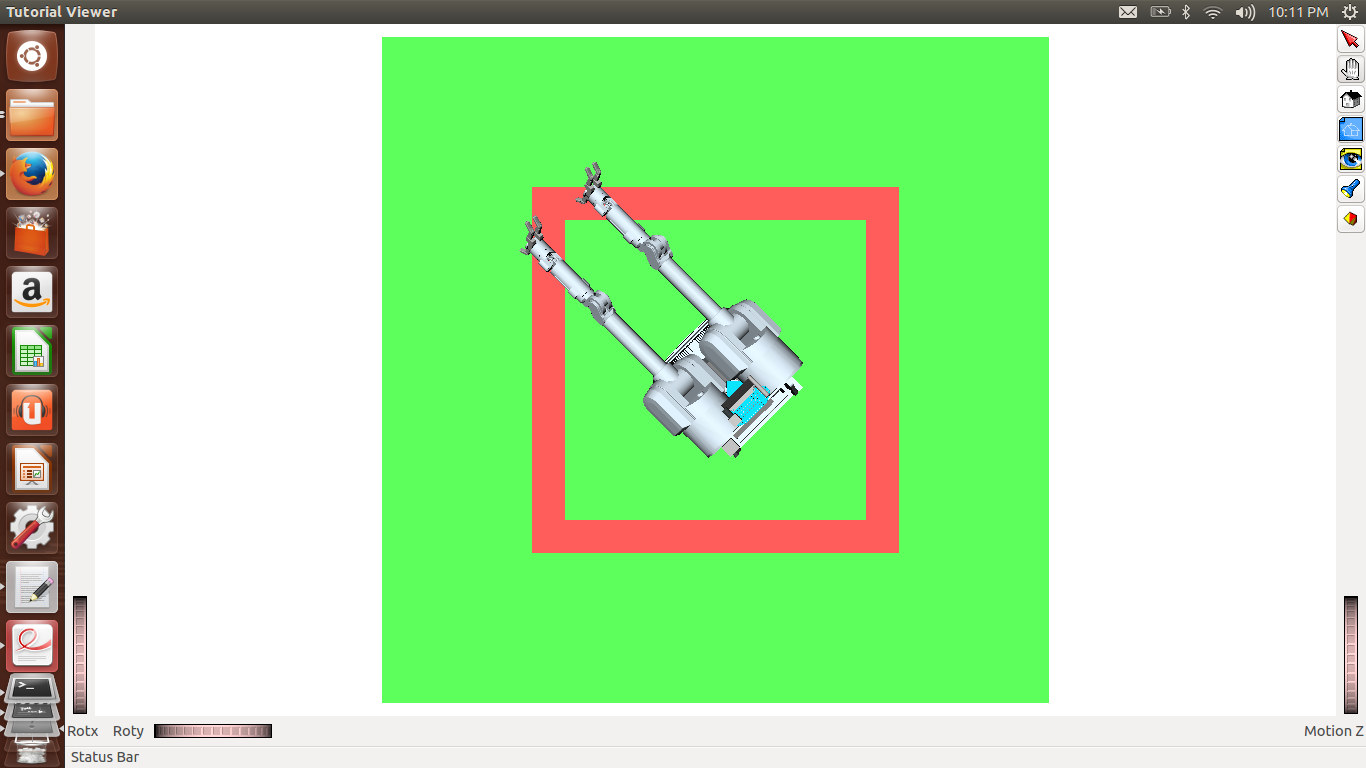
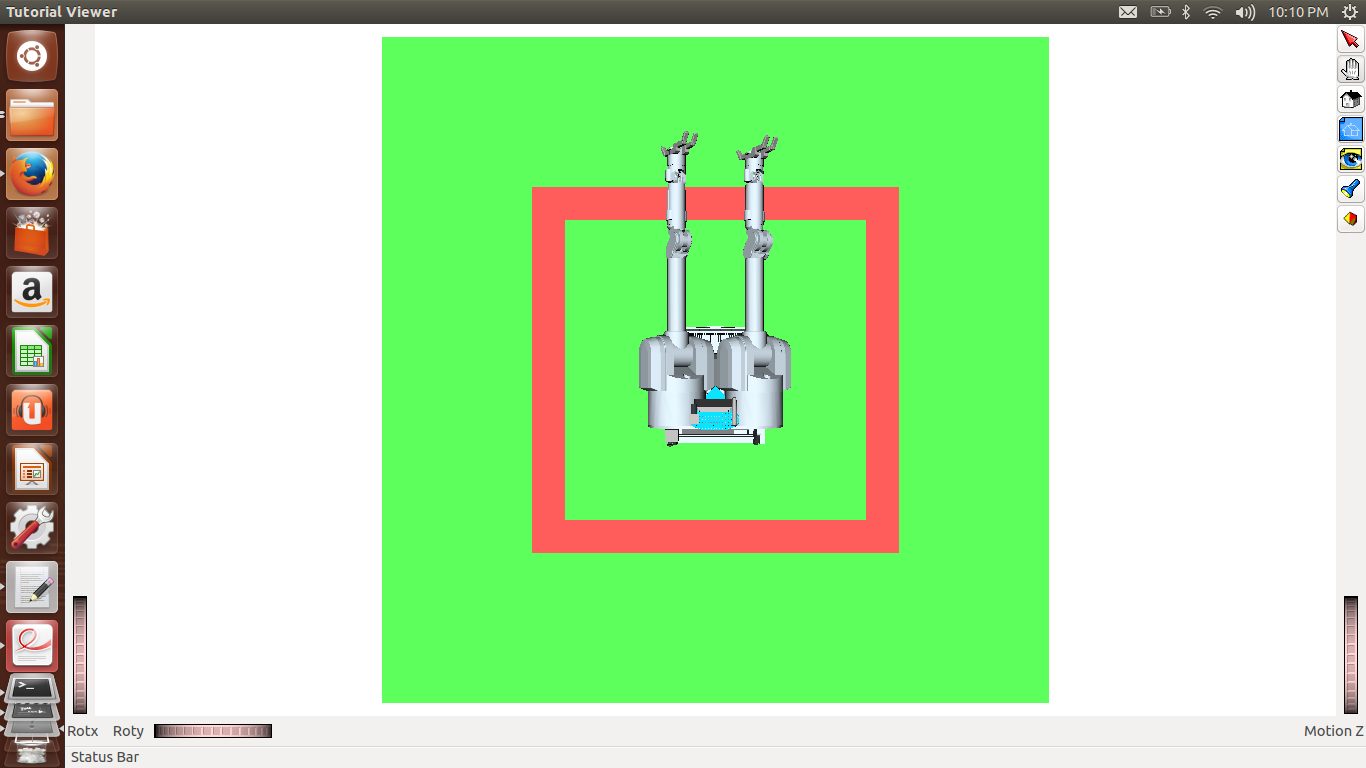
1.

2.

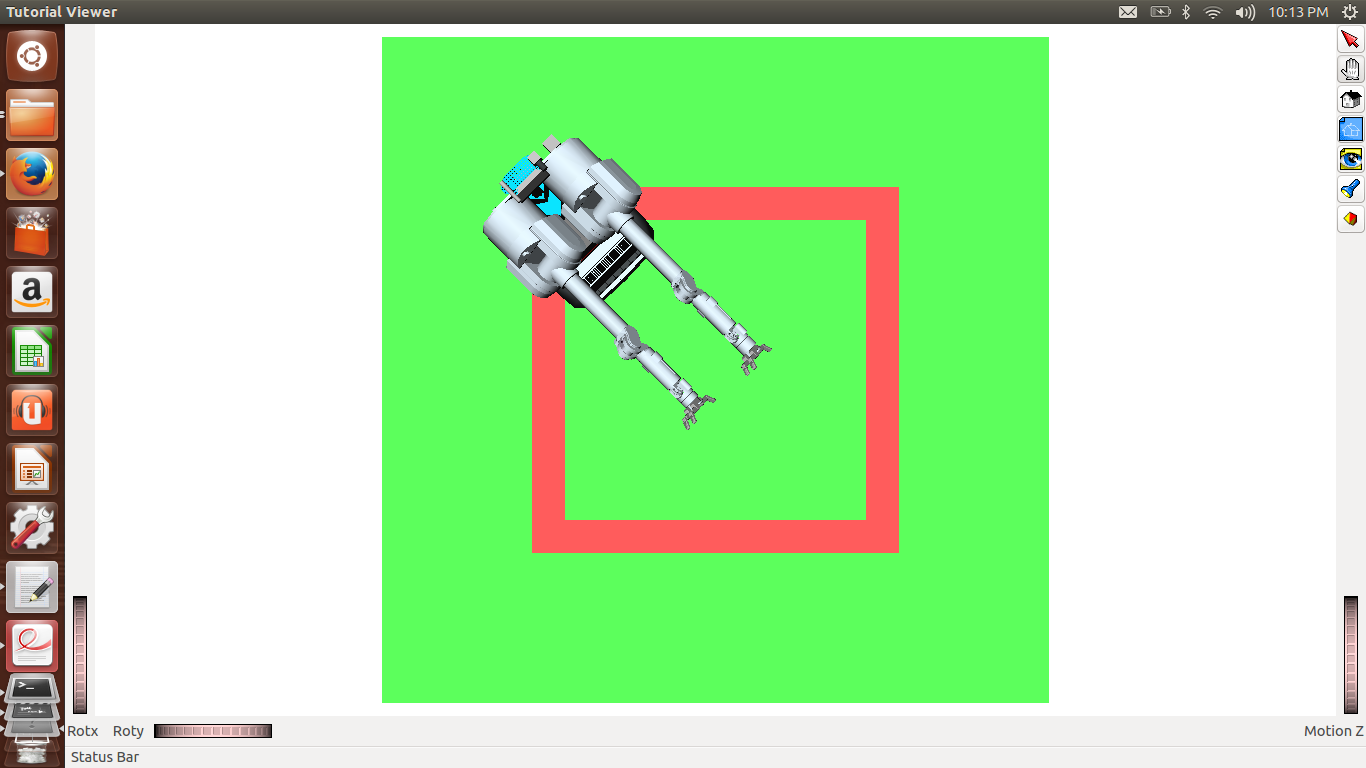
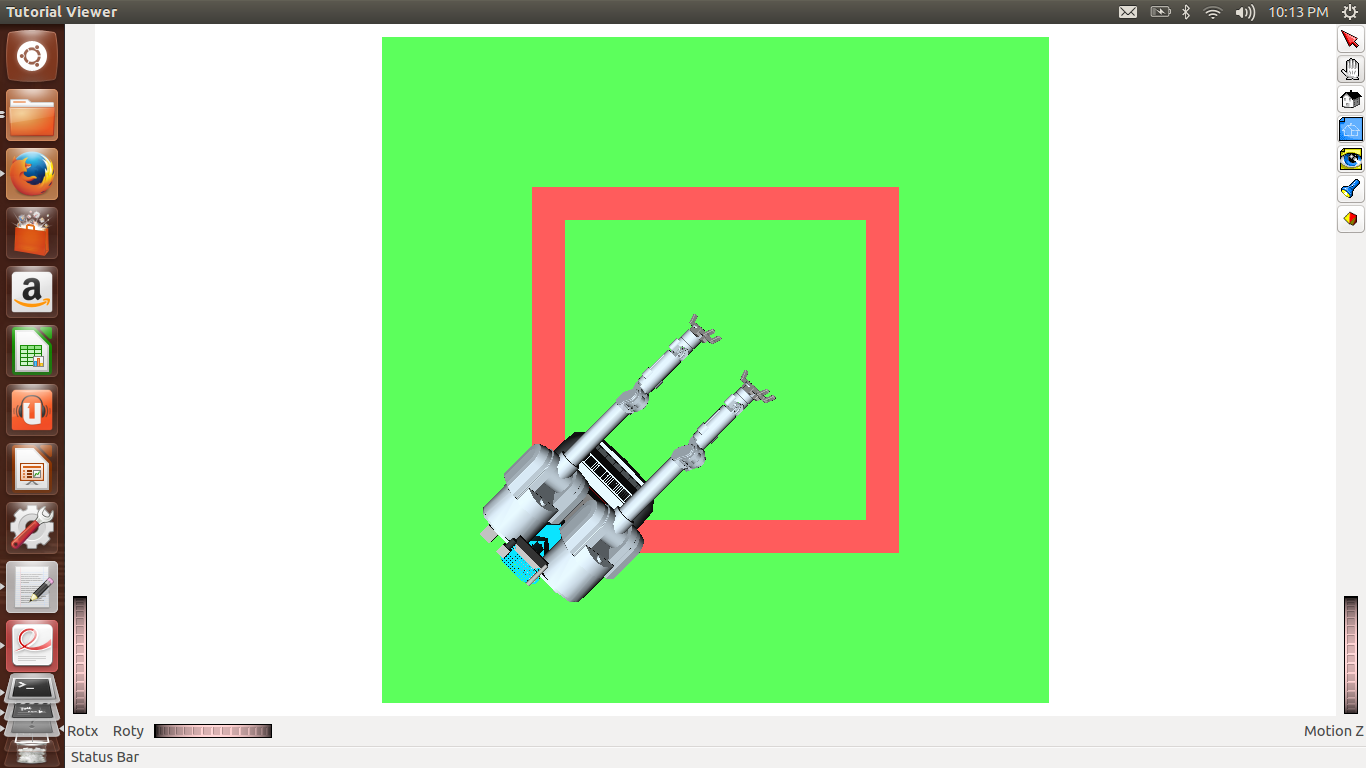
**2. rotate\_by**  
a) Code:  
***T = openravepy.matrixFromAxisAngle([0,0,ang]) # Get the transformation matrix  
with self.env:  
 self.robot.SetTransform(np.dot(T,self.robot.GetTransform())) # Applying the new #transformation***

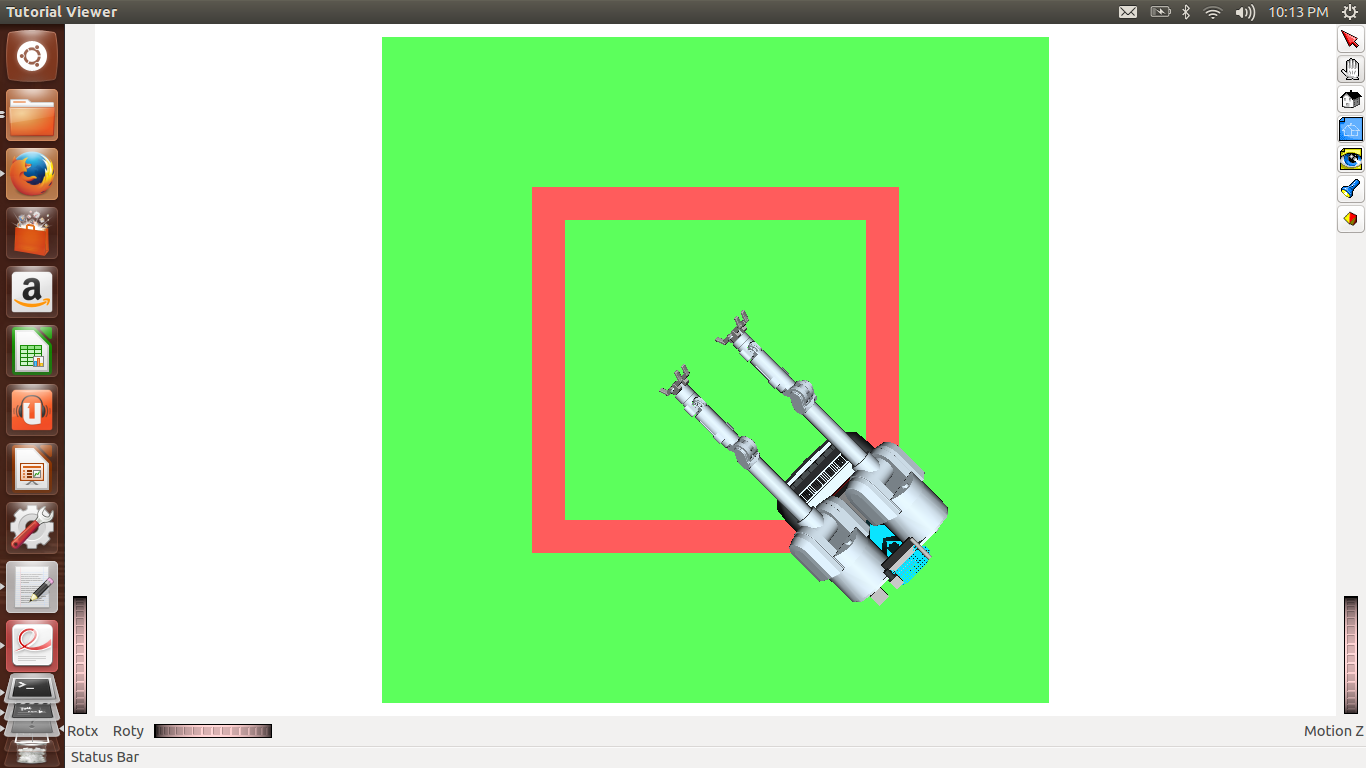
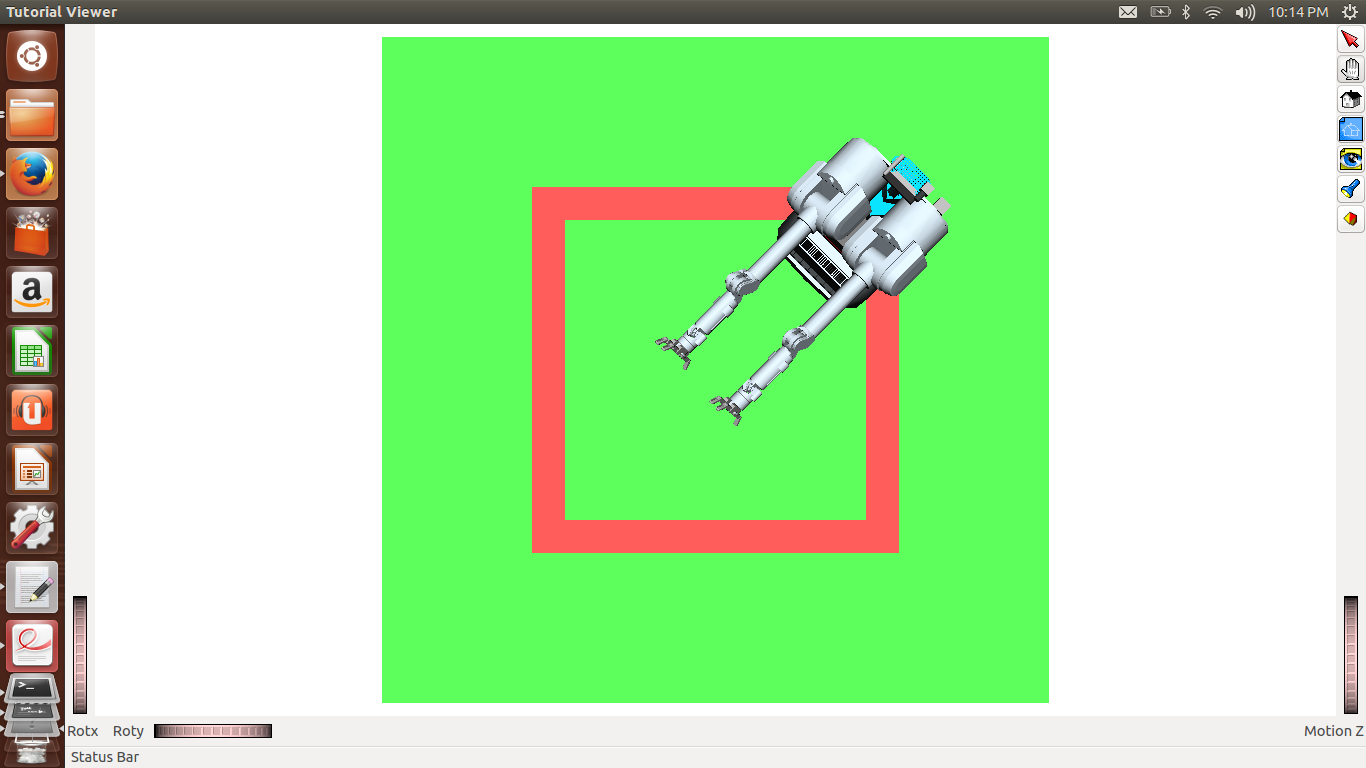
b) Images:  


**3.** **go\_around\_square**  
a) Code:

***# set the robot back to the initialize position after  
with self.env:  
 self.robot.SetTransform(np.identity(4));   
 self.rotate\_by(np.pi/4)  
 self.move\_straight(-np.sqrt(2))  
 time.sleep(3)  
 self.rotate\_by(np.pi/2)  
 time.sleep(3)  
 self.rotate\_by(np.pi/2)  
 time.sleep(3)  
 self.rotate\_by(np.pi/2)***

b) Images:

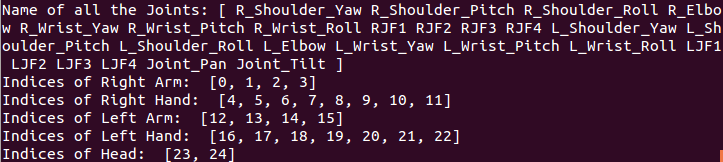
 

**4.** **figure\_out\_DOFS:**  
a) Code:

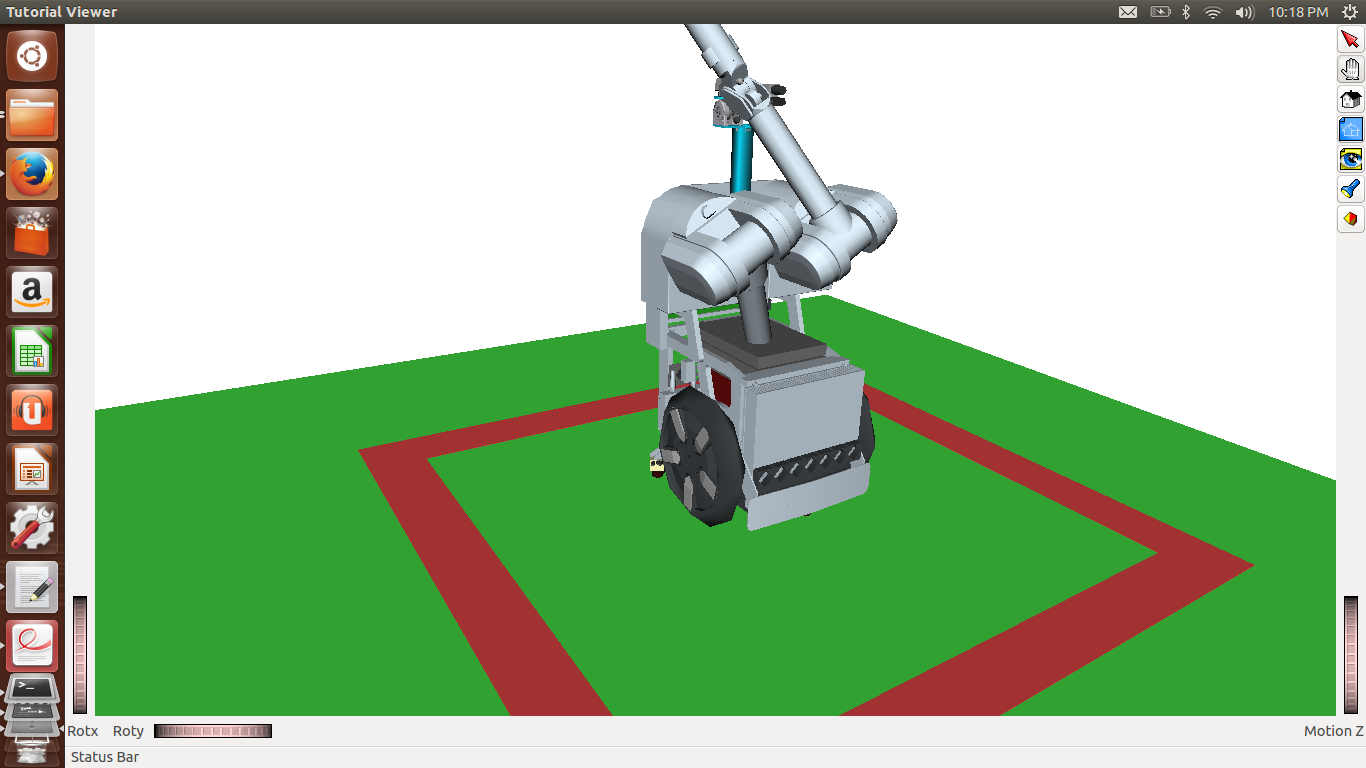
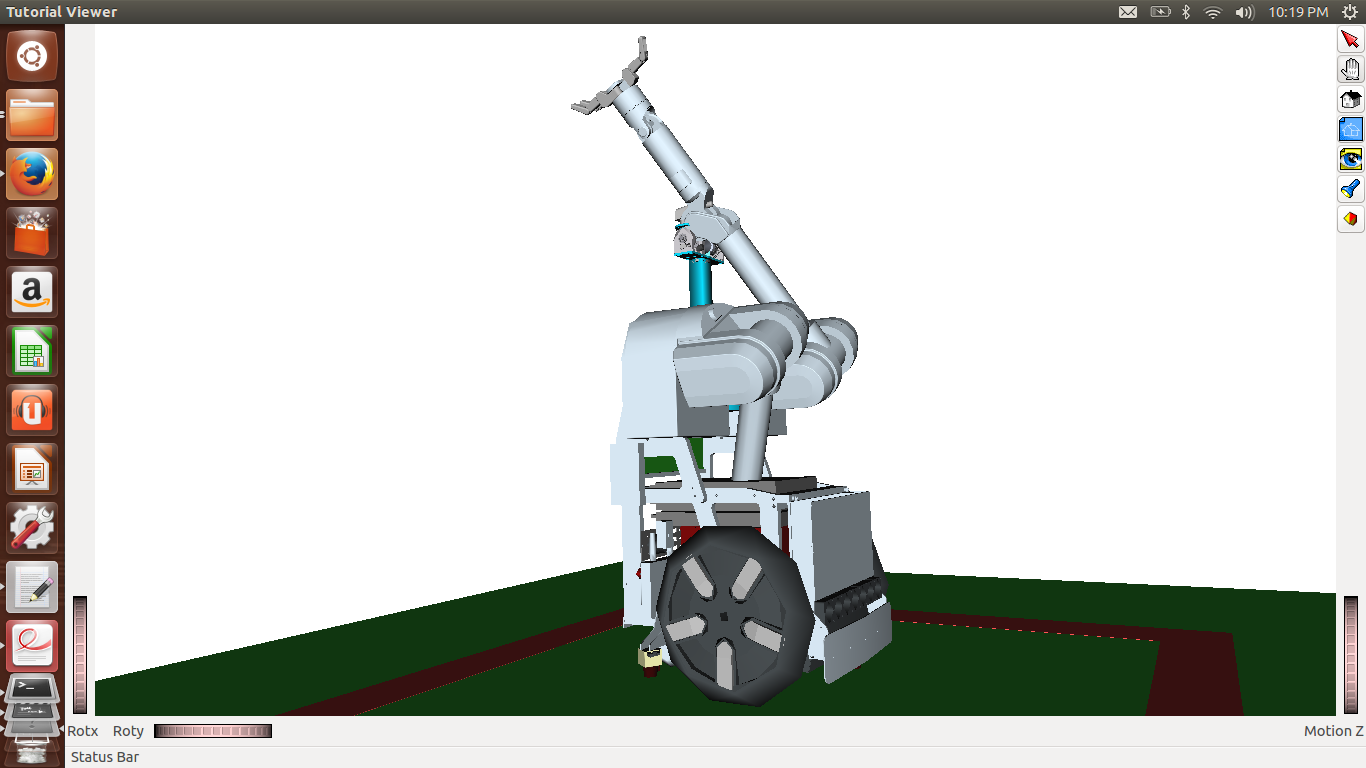
***### Joints  
with self.env:  
 Joint = self.robot.GetJoints()  
 print 'Name of all the Joints: [',  
 for i in xrange(0,len(Joint)):  
 print Joint[i].GetName(),   
 print ']'  
###  
### Indices  
print 'Indices of Right Arm: ',list([0,1,2,3])  
print 'Indices of Right Hand: ',list([4,5,6,7,8,9,10,11])  
print 'Indices of Left Arm: ',list([12,13,14,15])  
print 'Indices of Left Hand: ',list([16,17,18,19,20,21,22])  
print 'Indices of Head: ',list([23,24])  
###***

b) Name of Joints and Index Ranges (Note: Considering wrist and fingers as part of hand and elbows, shoulder as part of arm):



**5.** **put\_in\_self\_collision:**  
a) Code:  
***DOFValues = self.robot.GetDOFValues() # Get current DOF values  
DOFValues[1] = -3 # Modify one of them so that there is self collision  
DOFValues[12] = 3 # Modify another value  
Indices = xrange(0,len(DOFValues))  
with self.env:  
 self.robot.SetDOFValues(DOFValues,Indices,checklimits = False) # Set the new DOF #values and set checklimits = False***

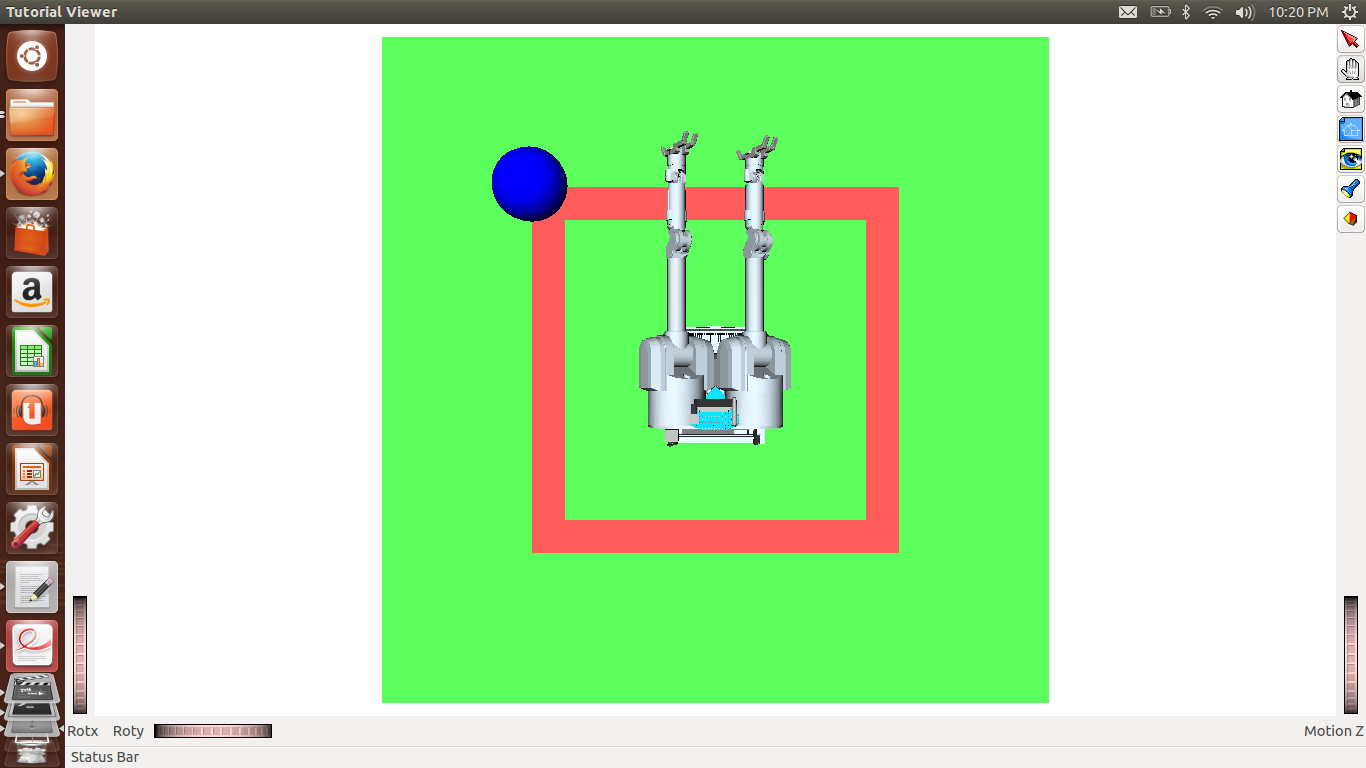
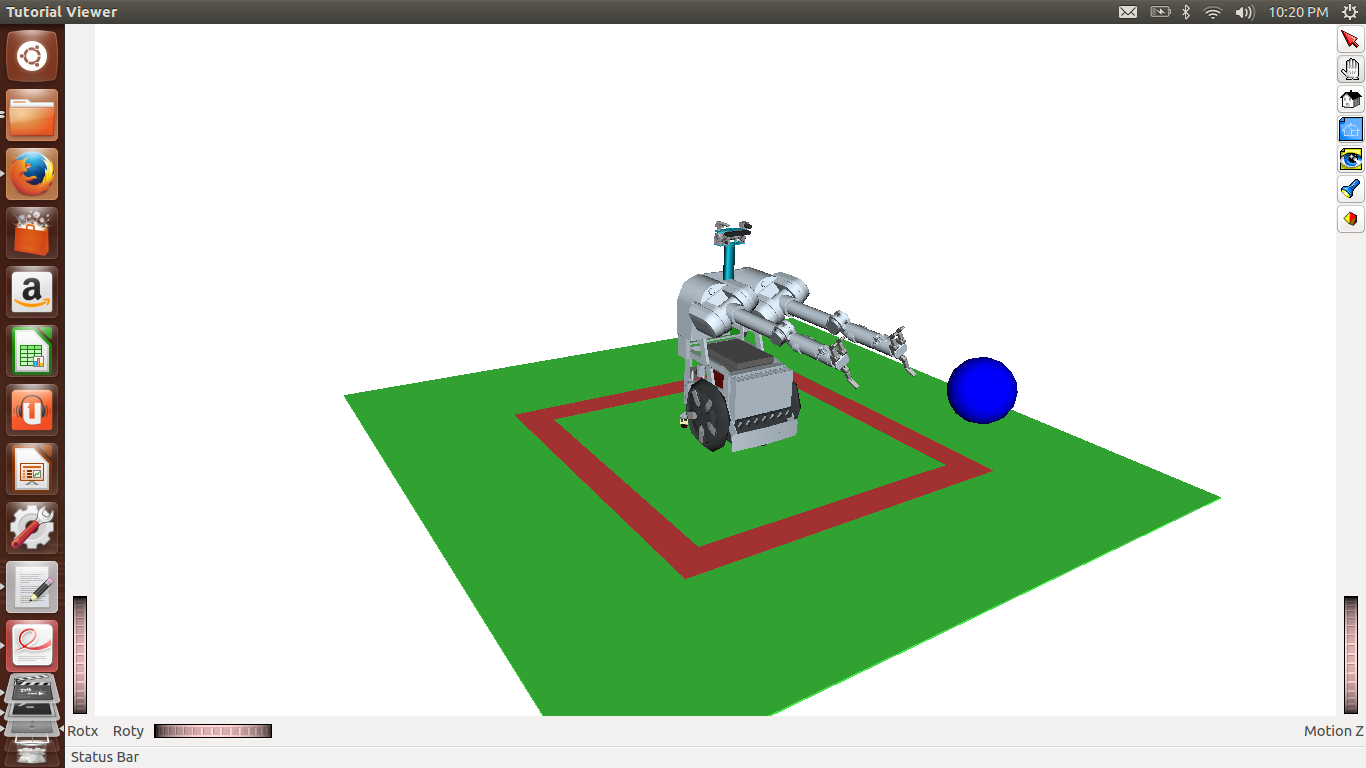
b) Images:

**6. Adding a Sphere**  
a) Code:

***<KinBody name="sphere1">  
 <Body type="static">  
 <Translation>1 1 0.5</Translation>   
 <Geom type="sphere">  
 <Radius>0.2</Radius>  
 <diffuseColor>0 0 1</diffuseColor>  
 <ambientColor>0 0 1</ambientColor>   
 </Geom>  
 </Body>  
</KinBody>***

b) Images:

**Time taken for this homework:**  
1. The first two python sorting functions did not take much time but for the third sorting function, I had to spend some time to figure it out. In total, Python sorting functions took around 1 hour.

2. OpenRave part of the assignment took some time. As this was the first time I was using OpenRave, I had to go through the documentation of OpenRave and spend some time to get familiar with it. Then, I started with the assignment. In total, it must have taken around 1 day to finish the OpenRave Assignment.

**Locking the environment:**  
To lock the environment, I have used ***with self.env*** in all the functions which locks the environment in python.