Lab Number 01
CoppeliaSim EDU – Introduction to the Software

Introduction:

The robotics simulator CoppeliaSim, with integrated development environment, is based on a distributed control architecture: each object/model can be individually controlled via an embedded script, a plugin, ROS / ROS2 nodes, remote API clients, or a custom solution. This makes CoppeliaSim very versatile and ideal for multi-robot applications. Controllers can be written in C/C++, Python, Java, Lua, MATLAB or Octave. Following are just a few of CoppeliaSim's applications:

- Simulation of factory automation systems
- Remote monitoring
- Hardware control
- Fast prototyping and verification
- Safety monitoring
- Fast algorithm development
- Robotics related education
- Product presentation

Lab Prepared by:
Dr. Tayyab Zafar
LE Hamza Sohail

Software Used:

CoppeliaSim EDU v4.7

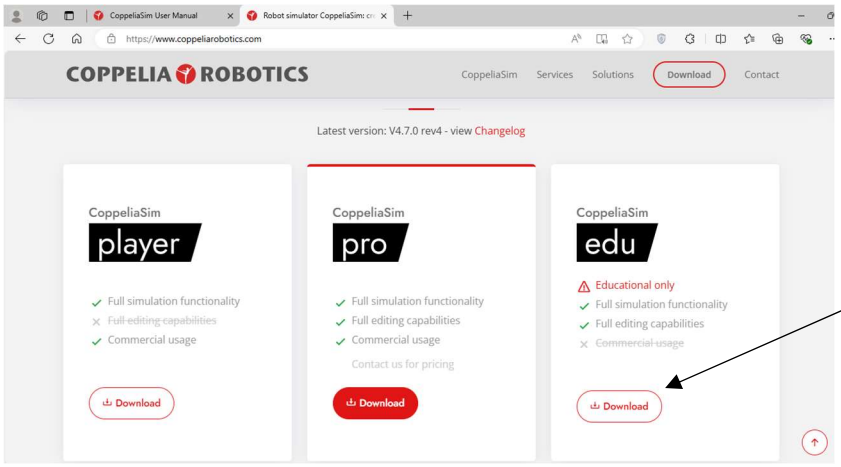
Objectives of the Lab:

Objectives of the Lab are:

- Downloading of CoppeliaSim EDU Software
- To explore capabilities of CoppeliaSim EDU Software

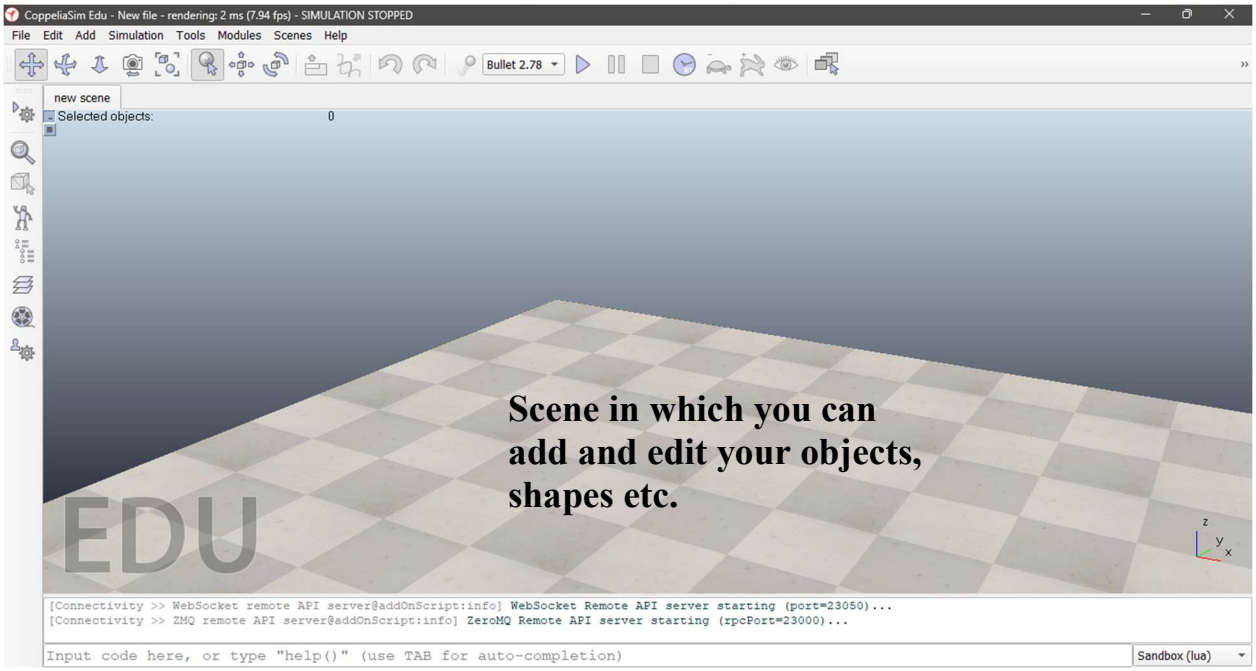
Downloading:

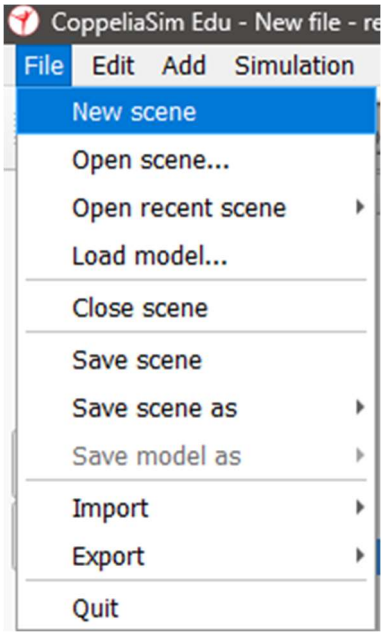
CoppeliaSim EDU is an open-source software. It can be downloaded from the website www.coppeliarobotics.com



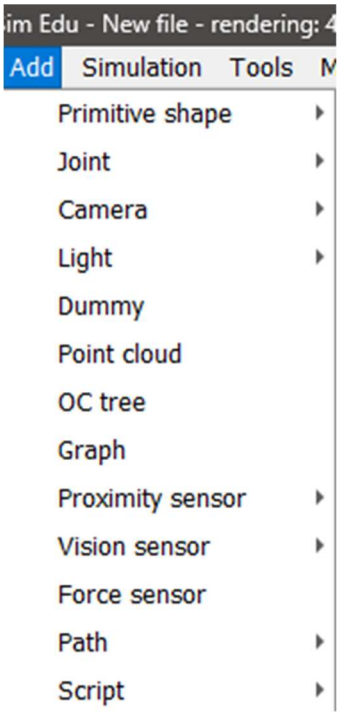
Download the EDU version. It is open source.

Upon opening CoppeliaSim EDU software we this window as follows:





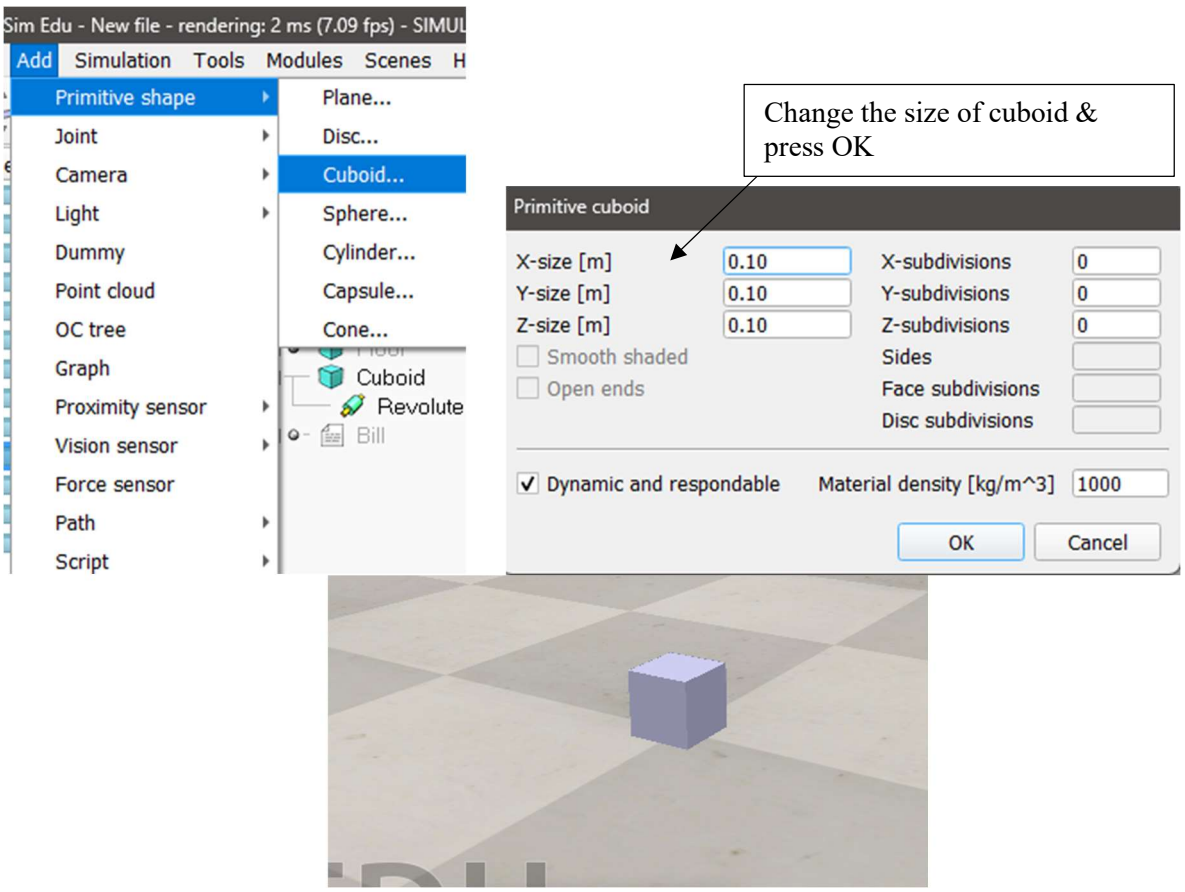
New Scene – add a new scene in which you can simulate
Open Scene – Open already existing scene
Close Scene – Close the scene already selected
Save Scene – Save the current scene
Quit – Exit the CoppeliaSim



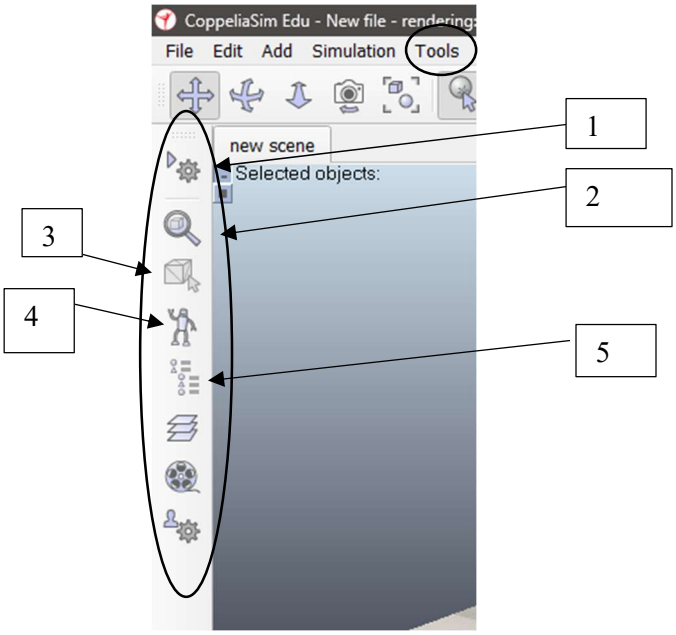
You can add shapes (cuboids etc), Joints (revolute, prismatic or spherical), Sensors or Script to

the scene.

Adding a Cuboid:

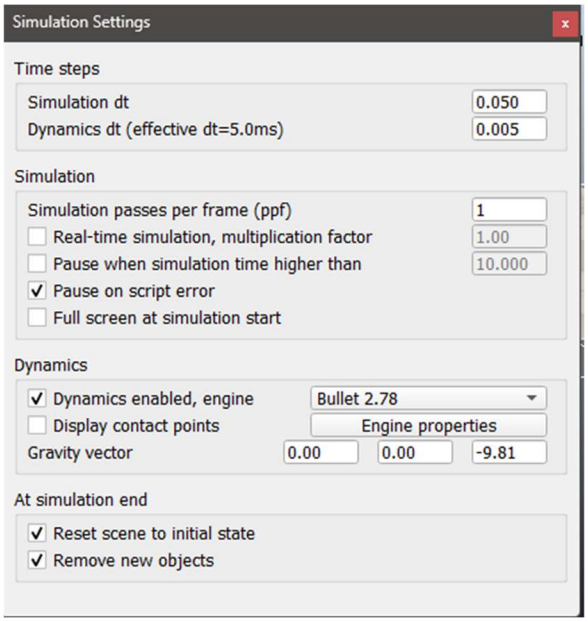


Tools are shown below:



1- Simulation Settings

In simulation settings dynamics should be enabled. At simulation end shows if you want to re initialize the scene from its initial state before it was simulated. This is helpful if you want to add new objects to want to check position and orientation of objects during simulation.

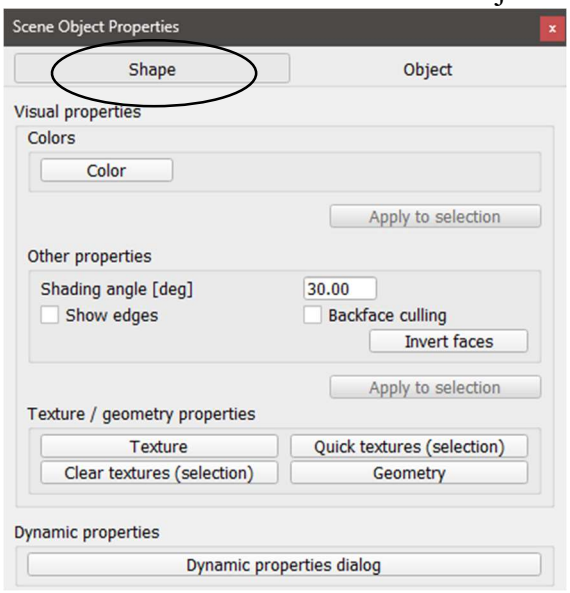


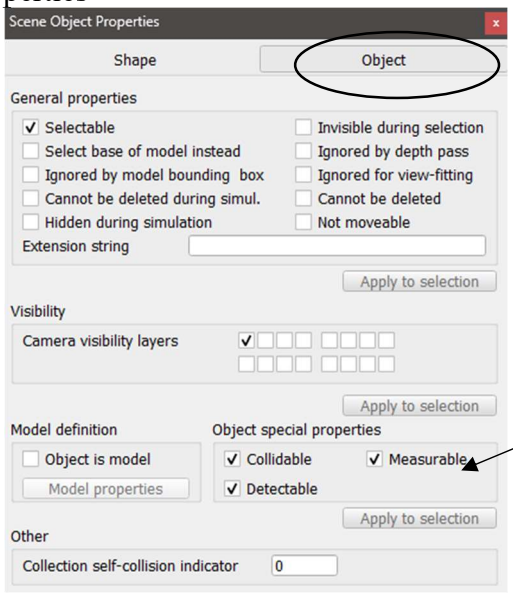
2- Scene Object properties

Once you select an object and click on scene object properties then a window will

appear in which you can change properties as per requirement.

Select Cuboid
Click on the "Scene Object Properties"



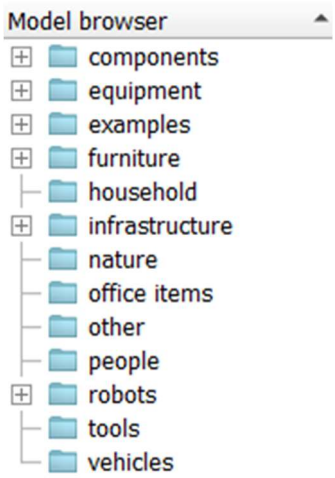


To set an object as either collidable, measurable or detectable. Such as Force sensor can only sense collidable object

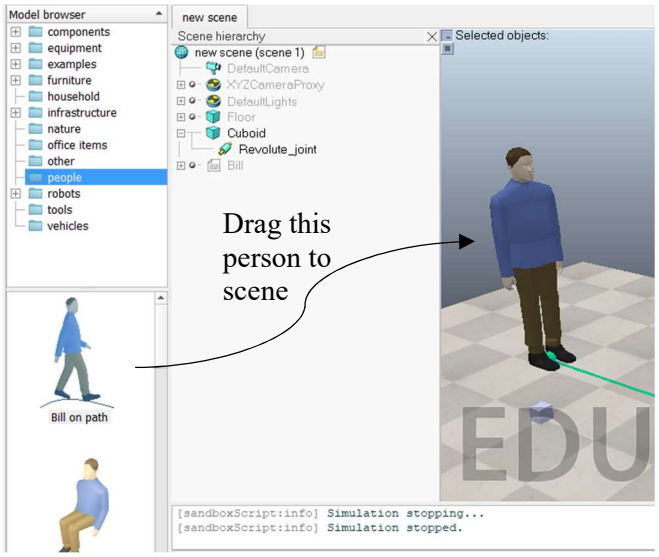
3- Shape Edit Mode

This option enables you to extract one shape from an already existing object.

4- Model Browser



Model browser shows objects, equipment or robots that can be placed inside the scene. In this browser go to people and drag one to the scene.



5- Scene Hierarchy

Scene Hierarchy shows parent and child objects.

Add a cuboid
Add a revolute joint
Add another cuboid

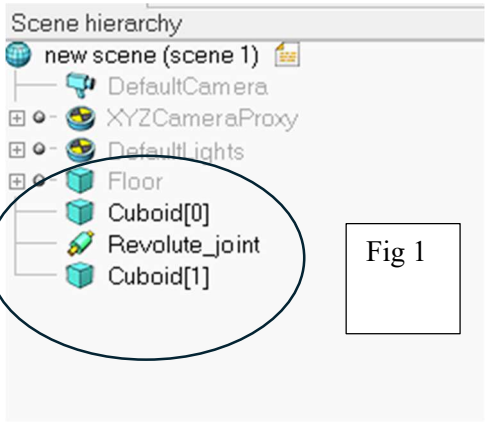


Fig 1

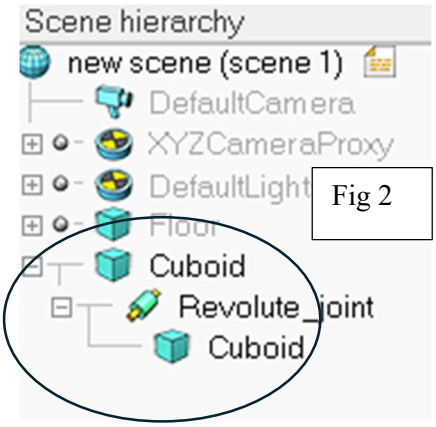


Fig 2

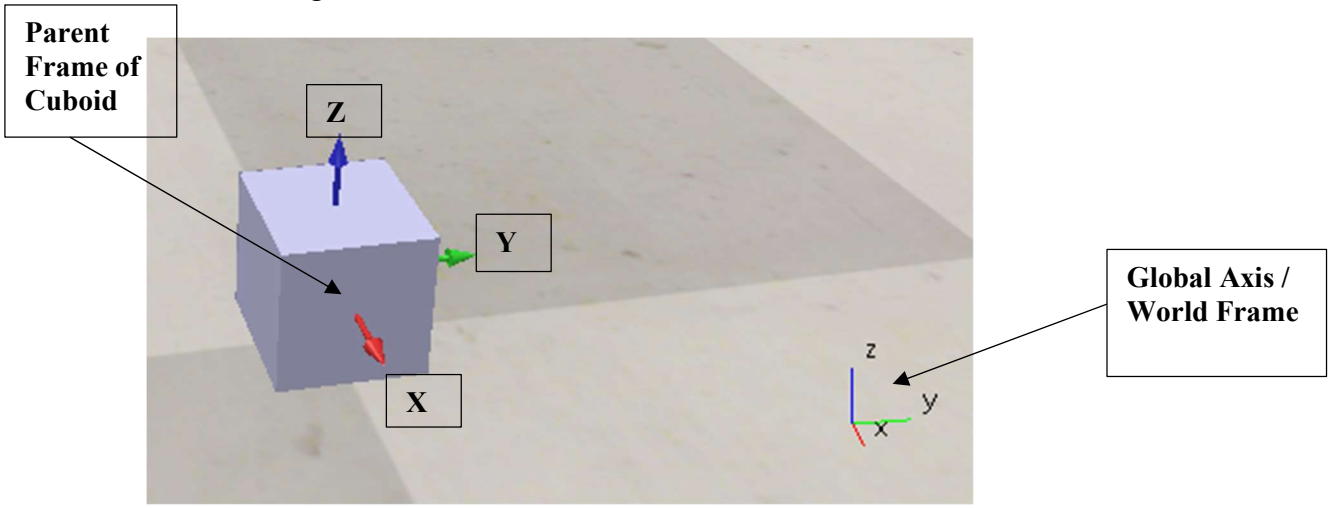
Figure 1 shows independent objects on the scene such that revolute joint is not a child and cuboid[0] is not responsible for revolute joint. In figure 2 Revolute joint is a child of Cuboid parent. And cuboid2 is a child of Revolute joint such that revolute joint is connected to Cuboid parent and cuboid2 is connected with revolute joint. Here Revolute joint is responsible for child cuboid to move.

DUMMY Objects:

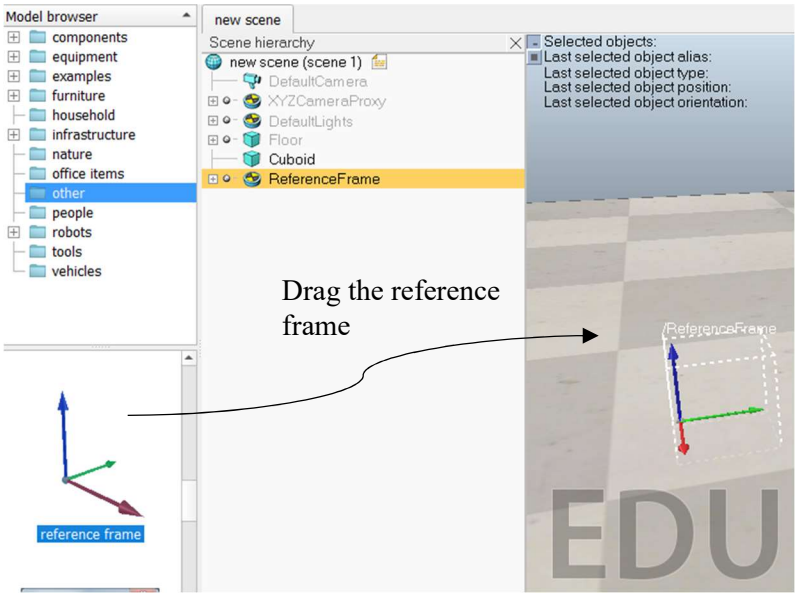
Dummy Objects does nothing in the scene but provide assisting in coding as reference to an object.

Translating and Rotating an Object/Frame

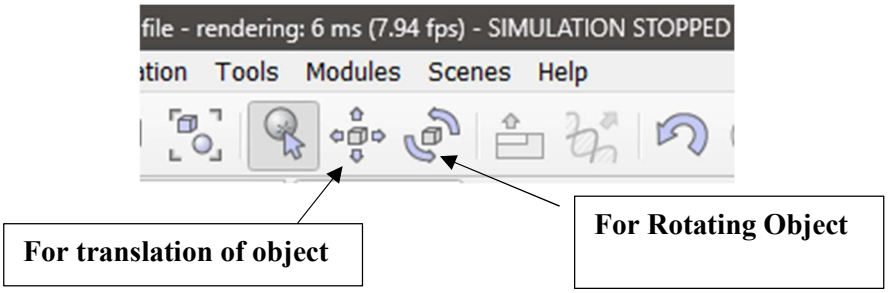
Translating means moving Objects/Frames from one place to another upon X, Y or Z axis while rotating means rotation of axis around X, Y or Z axis.



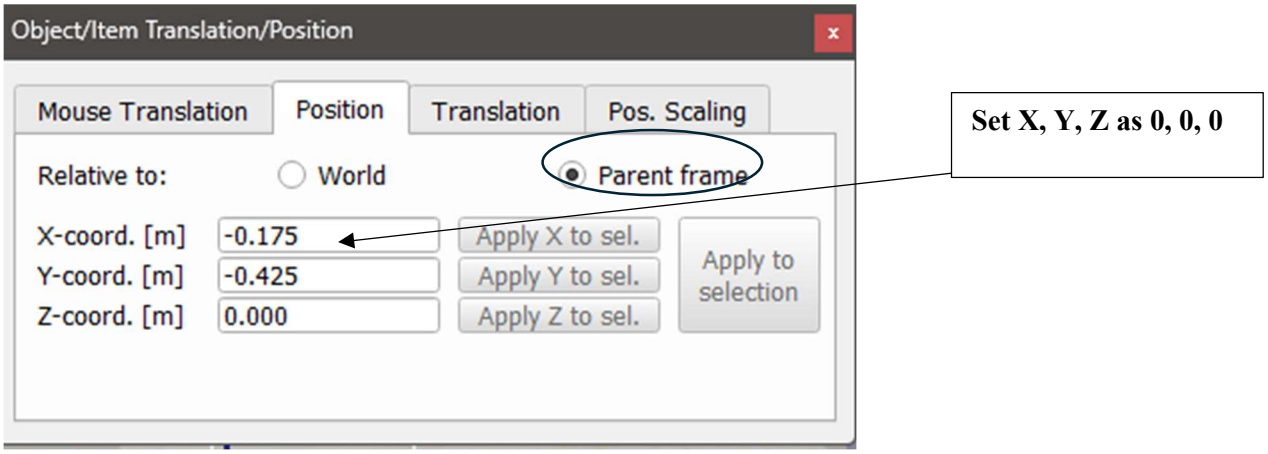
Object can be translated and rotated either by parent frame or world frame. In model browser, got to others → Drag reference frame to scene.



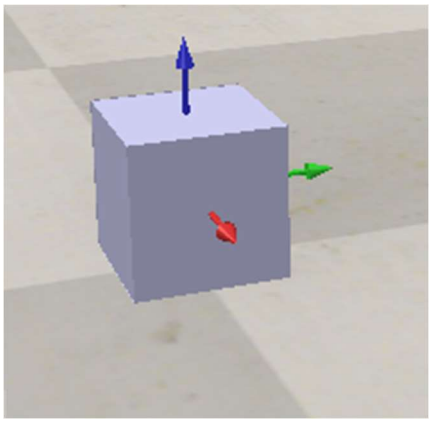
Attach it to the cuboid. Drag the reference frame in scene hierarchy to cuboid.



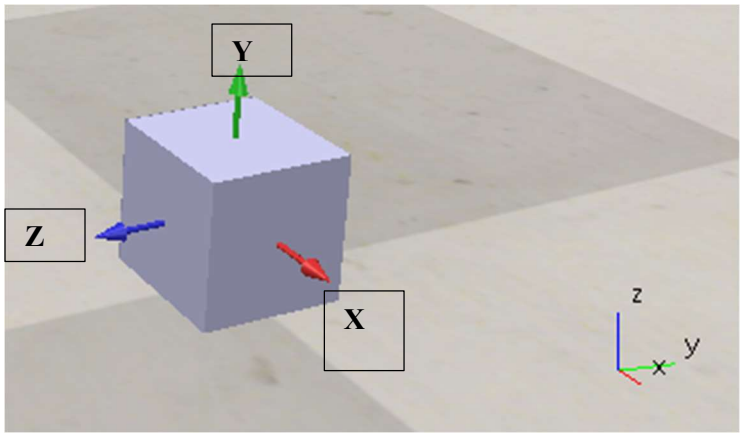
Select the reference frame and click on the translational settings.



To attach it to the reference frame, select parent frame 0, 0 ,0 such that it will translate to cuboid since it is a child object of cuboid. Also select rotation and make its parent frame 0,0,0.



Now select the cube and change its orientation (Alpha +90) relative world frame.



Now you can see world frame axis are different than the cuboid parent frame axis.