CUTeR Arm Simulator User Manual

1 Contents

2 Getting Started

2.1 Installation

1. Download the application from GitHub.
2. Install Python (ver. 3.0+).
3. Install PyCharm (Optional).

2.2 Start

2.2.1 Windows version

1. Start the trajectory server (Optional)
   a) You need to start the trajectory server for dynamic trajectory generation, you can skip this part if you don't use the dynamic trajectory generation. You can use PyCharm or terminal or any other method to execute the python file. We will introduce how to execute the python file using PyCharm or terminal on windows in this section.
   b) Using terminal

      First, you need to open terminal. There are several ways to open terminal. You can search them online, here I just describe one. Press 'Windows key' > Type 'cmd' > Press 'Enter'.

      After having the terminal open, you need to change your working directory to the application root folder. You can use 'cd <Path to root folder>' to change your working directory. Finally, type 'python3 StartServer.py' then press enter to run the trajectory server.
   c) Using PyCharm.

      First run PyCharm. Open the application's folder as a project. Add an interpreter if needed. Right click 'StartServer.py' on the left-handed window. Click 'Run 'StartServer.py'.
2. Start the simulator
   a) Double click 'CUTeR ARM Simulator.exe' on the application folder to start the simulator.

2.2.2 Web version

   b) First, you need to start the simulator server. You can use PyCharm or terminal or any other method to execute the python file. We will introduce how to execute the python file using PyCharm or terminal on windows in this section.
   c) Using terminal

First, you need to open terminal. There are several ways to open terminal. You can search them online, here I just describe one. Press ‘Windows key’ > Type ‘cmd’ > Press ‘Enter’.

After having the terminal open, you need to change your working directory to the application root folder. You can use ‘cd <Path to root folder>’ to change your working directory. Finally, type ‘python3 StartServer.py’ then press enter to run the trajectory server.

d) Using PyCharm.

First run PyCharm. Open the application’s folder as a project. Add an interpreter if needed. Right click ‘StartServer.py’ on the left-handed window. Click ‘Run ‘StartServer.py’.

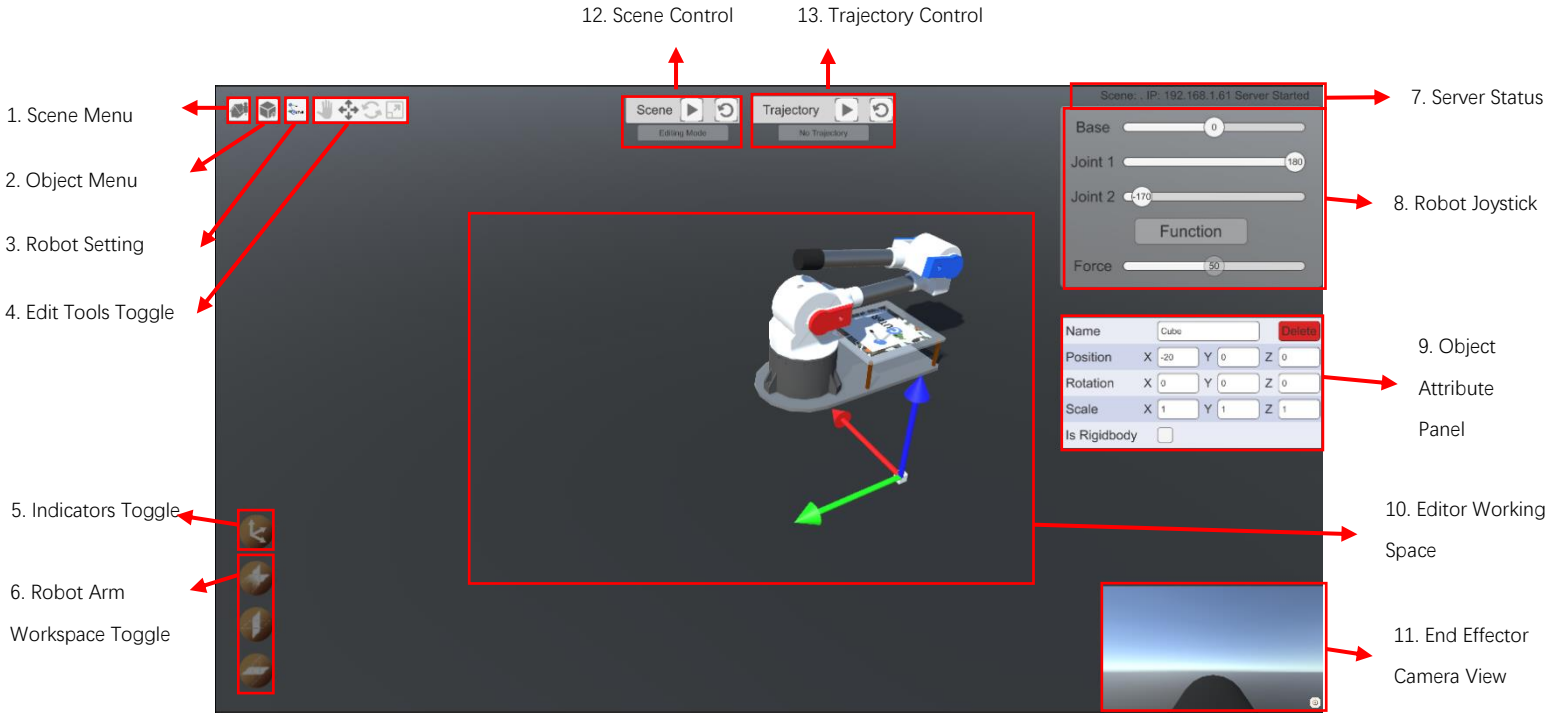
3. Start the simulator

a) Open any web browser. Then enter the address: ‘localhost:8000’ and press enter.

3 Using the application

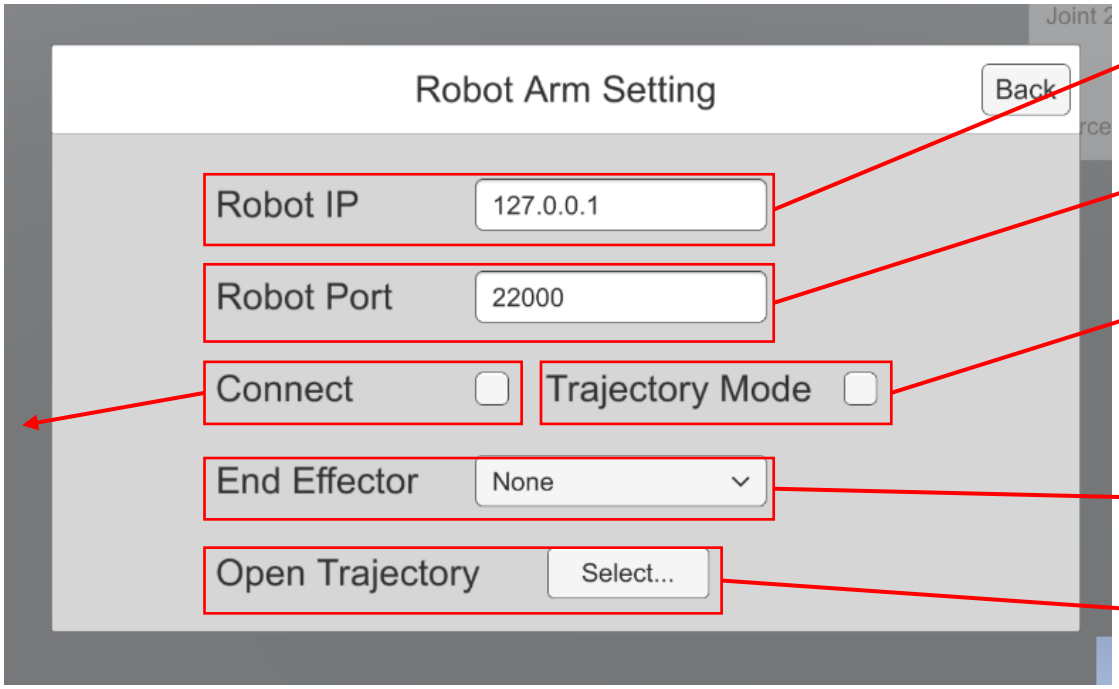
3.1 Overview

3.1.1 Main screen



Index	Name	Function
1	Scene Menu	Open scene menu.
2	Object Menu	Open object menu.
3	Robot Setting	Open robot setting panel.
4	Edit Tools Toggle	Choose edit tools by clicking correspond button.
5	Indicators Toggle	Turn on/off frames and indicators.
6	Robot Arm Workspace Toggle	Turn on/off visualization of robot arm's workspace.
7	Server Status	Information of simulator's status.
8	Robot Joystick	For controlling robot arm.
9	Object Attribute Panel	Edit selected object's attribute.
10	Editor Working Space	Main working space of the editor.
11	End Effector Camera View	View of end effector camera.
12	Scene Control	Scene mode status indicator and control buttons.
13	Trajectory Control	Trajectory status indicator and control buttons.

3.1.2 Robot Arm Setting Panel



1. Robot Server IP Configuration

2. Robot Server Port Configuration

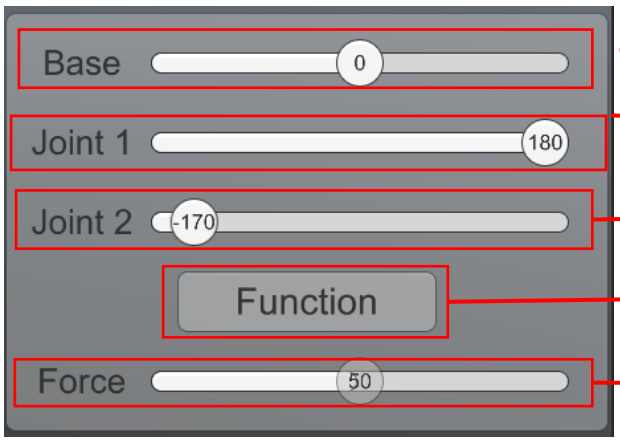
3. Connection Switch

4. Trajectory Mode Switch

5. End Effector Selector

6. Load Trajectory Button

3.1.3 Joystick Panel



1. Base Joint Angle Slider

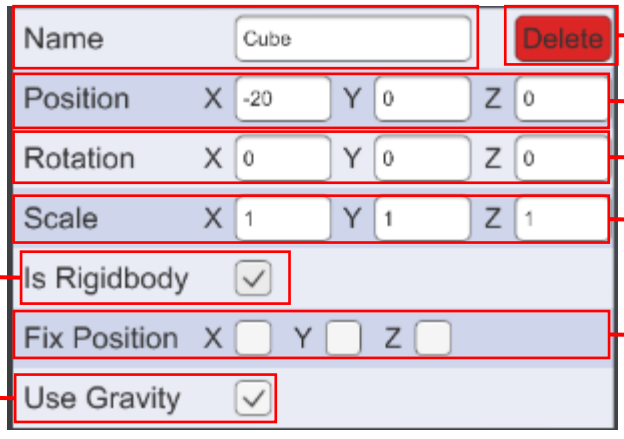
2. Joint 1 Angle Slider

3. Joint 2 Angle Slider

4. End Effector Fire

5. Launcher Force Slider

3.1.4 Attribute Panel



1. Delete Button

2. Object Position

3. Object Rotation

4. Object Scale

5. Rigid body Toggle

6. Fix Position Toggle

7. Gravity Toggle

3.2 Editor Viewport Navigation

3.2.1 Rotate

Hold right mouse button then move your mouse to rotate.

3.2.2 Move

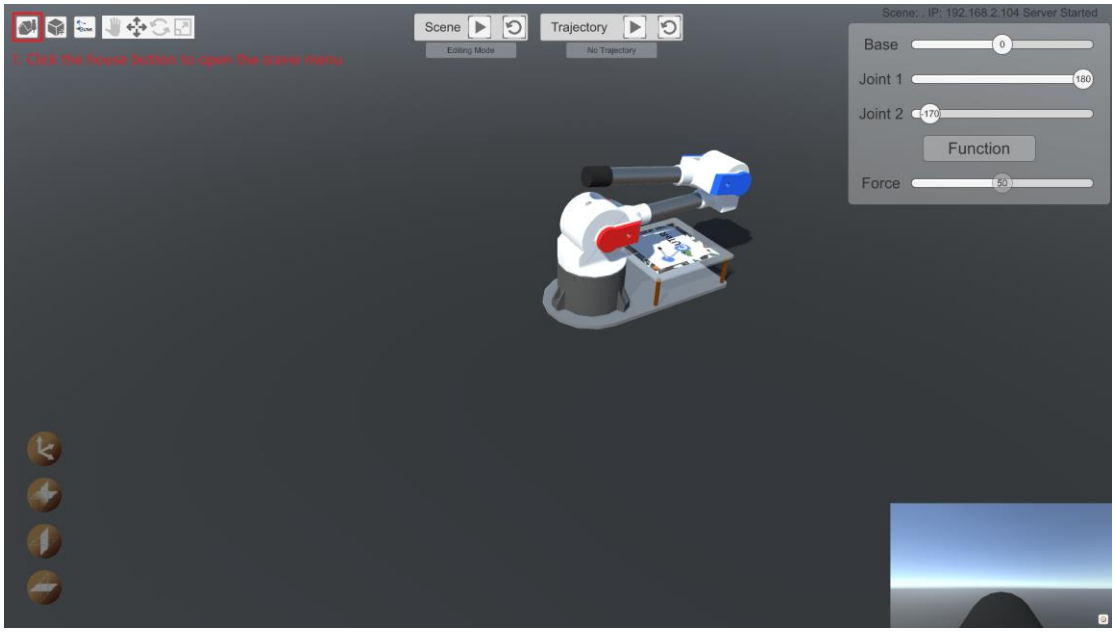
1. Hold middle mouse button then move your mouse to move.
2. Select the ‘Move Tool’ in the ‘Edit Tools Toggles’. Hold left mouse button then move your mouse to move.

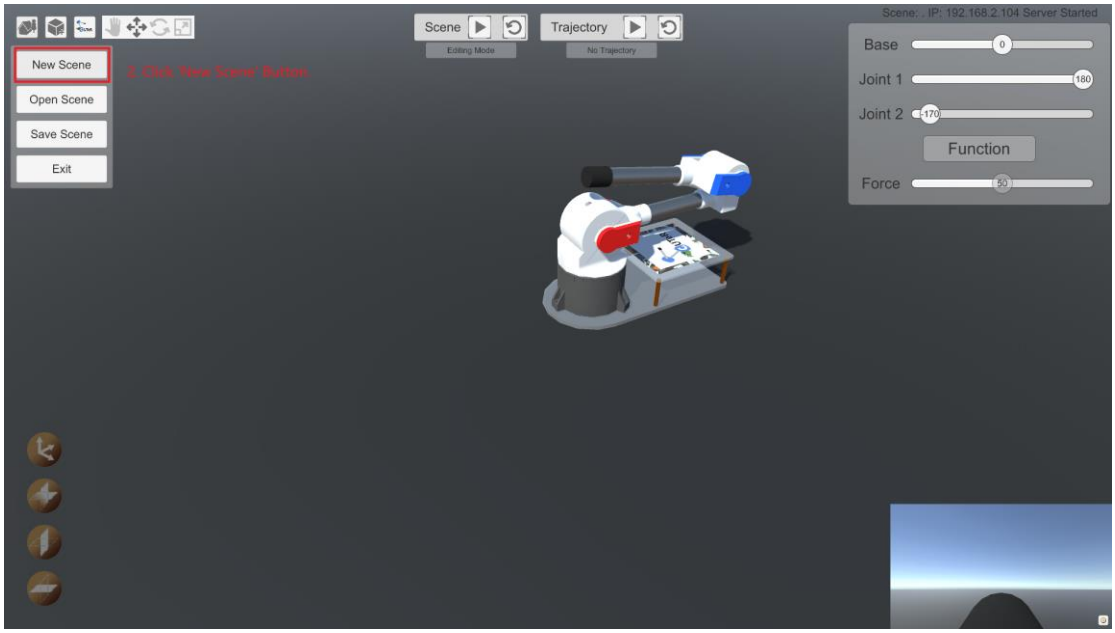
3.2.3 Zoom

Scroll your middle mouse button to zoom.

3.3 Creating a new virtual scene

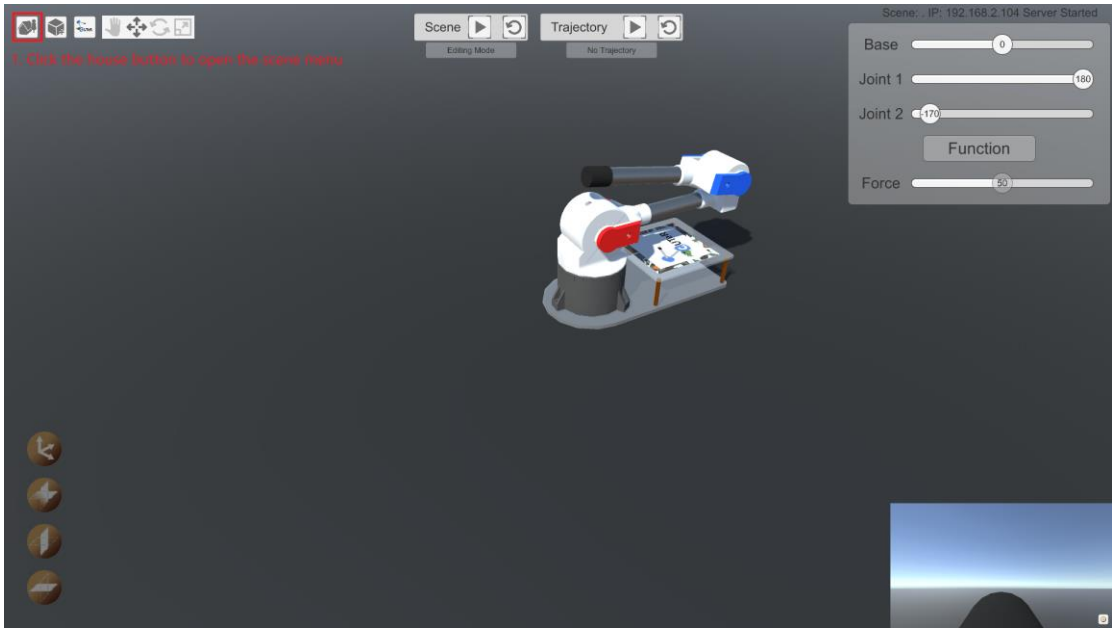
To create a new scene, select the ‘New Scene’ option from the scene menu.

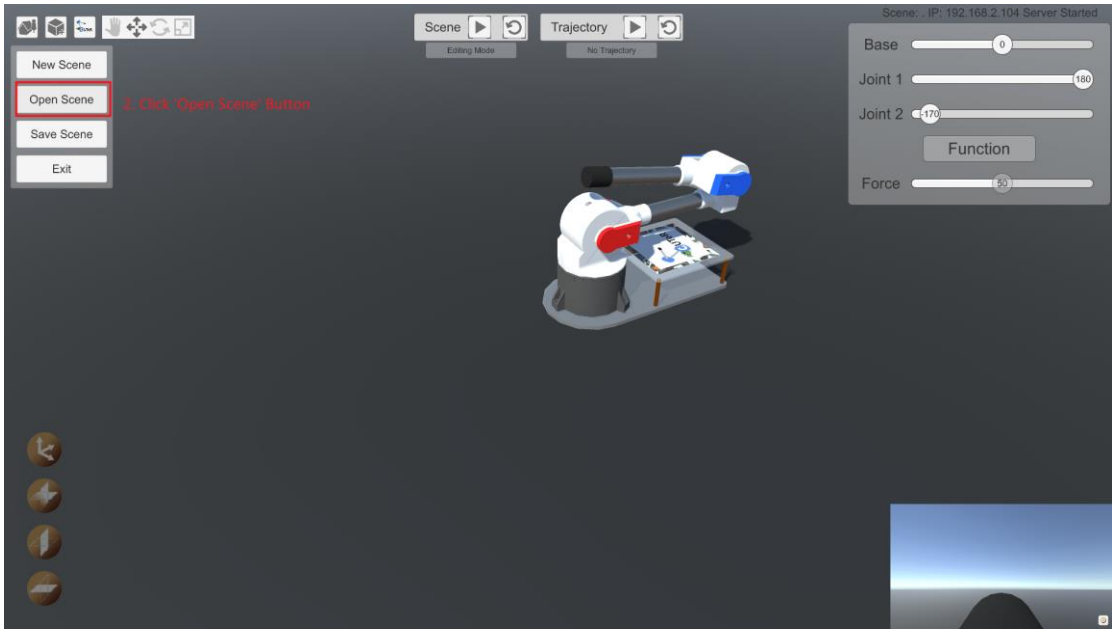


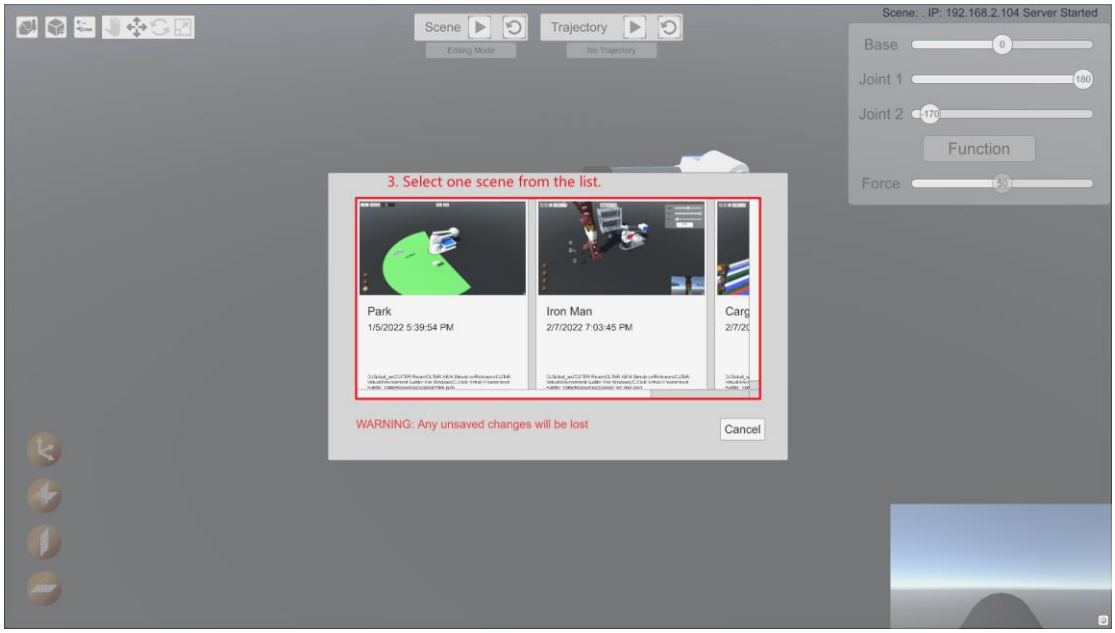

New Scene
Open Scene
Save Scene
Exit
1. Click 'New Scene' Button
Scene
Editing Mode
Trajectory
No Trajectory
Scene: . IP: 192.168.2.104 Server Started
Base
0
Joint 1
180
Joint 2
-170
Function
Force
50

3.4 Load an existing virtual scene

To load an existing scene, select the 'Load Scene' option from the scene menu. The load scene panel will pop up, you can then select one scene from the list.

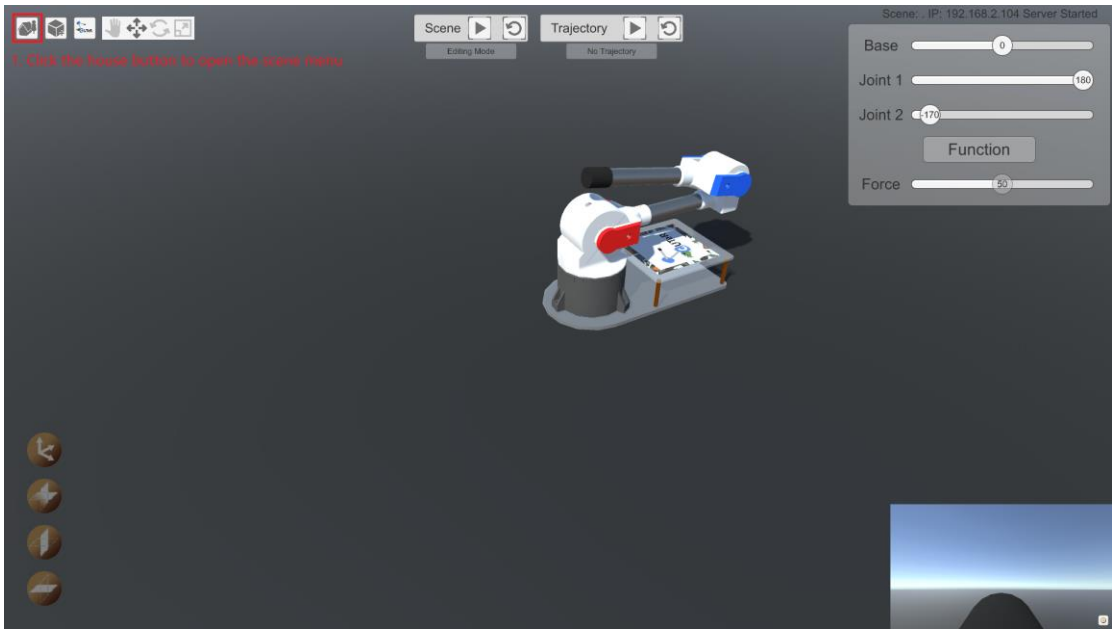


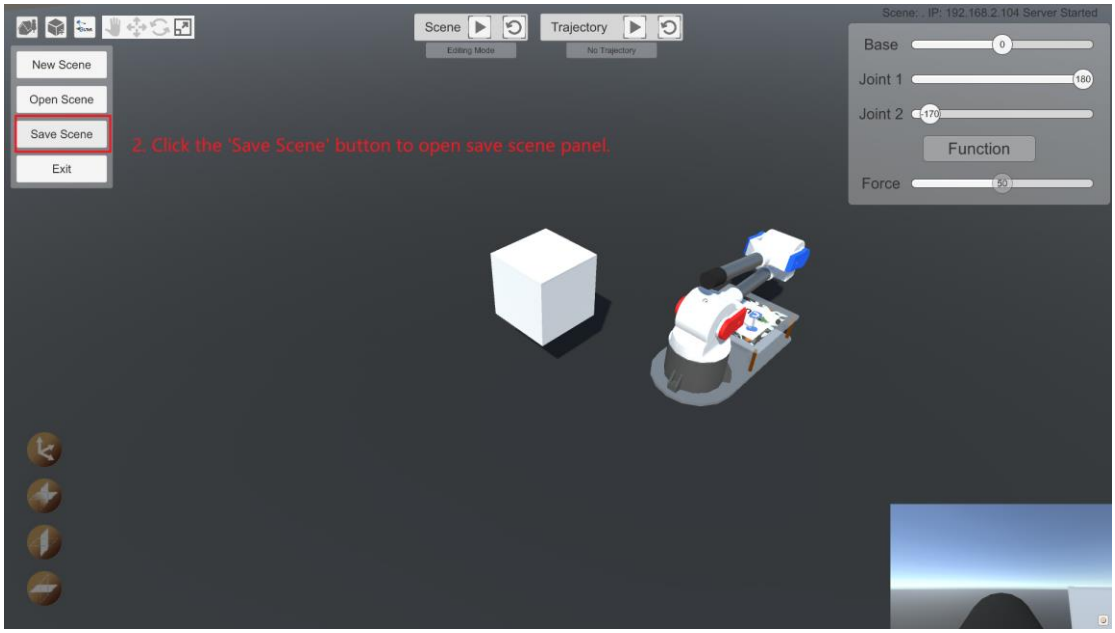




3.5 Save Current Scene

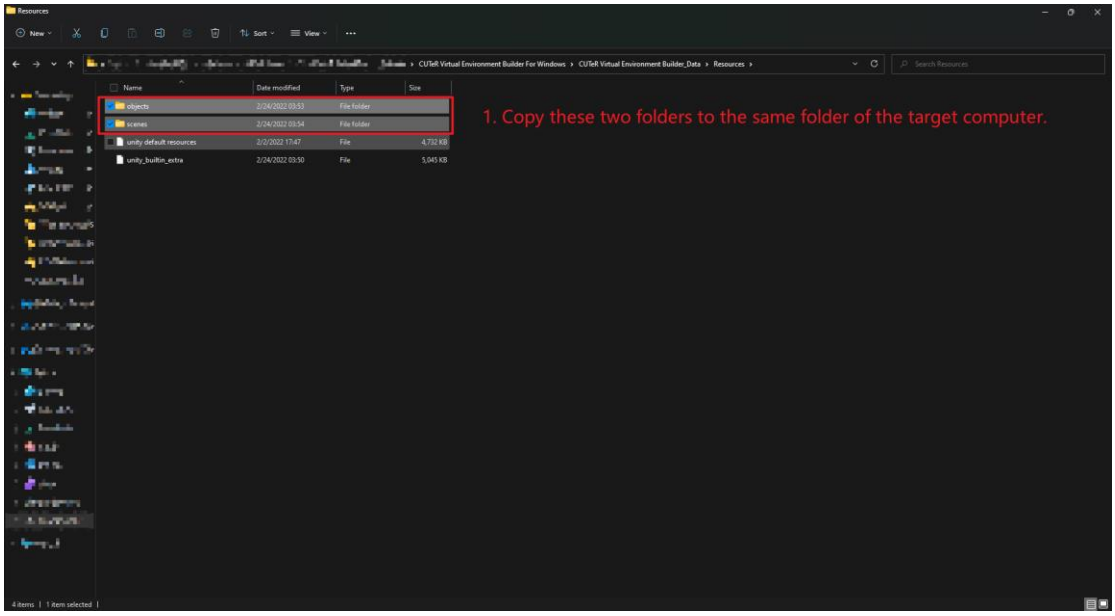
To save current scene, select the 'Save Scene' option from the scene menu. The save scene panel will pop up, fill in the name and description (optional), click save button.





3.6 Transfer Scenes

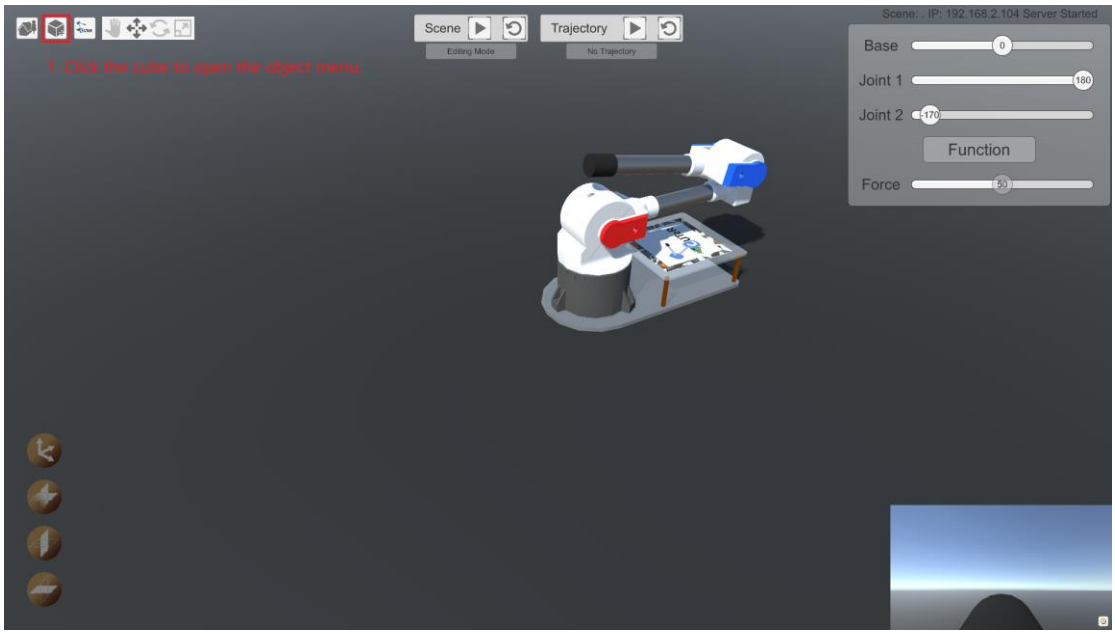
You may need to share your virtual scene with your classmates. To do so, you just need to copy the objects and the scenes folder to the corresponding directory of another computer.

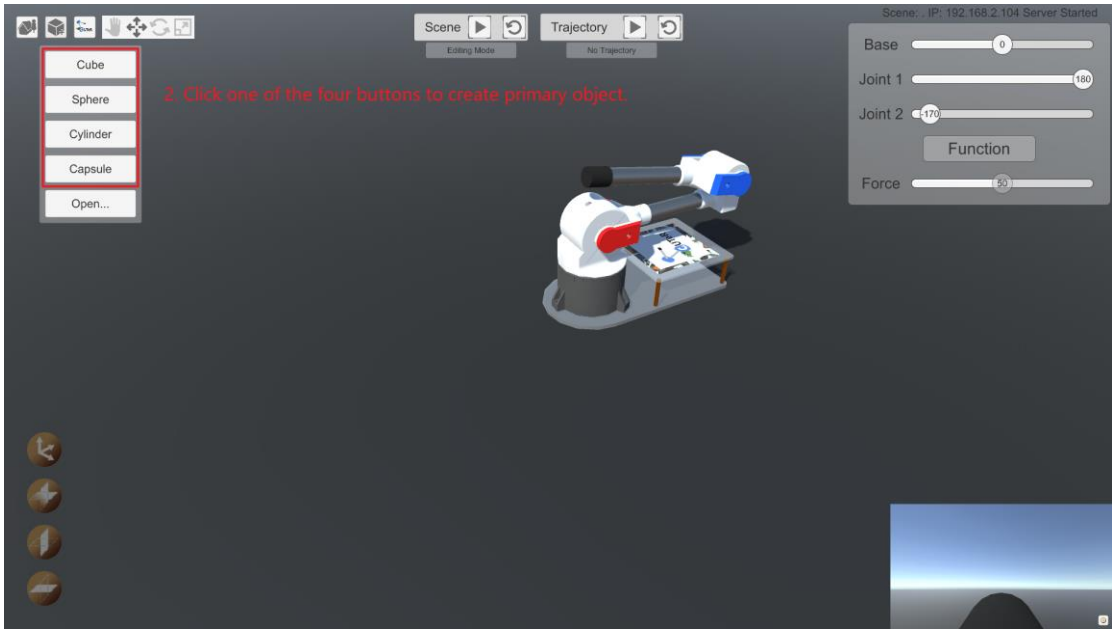


3.7 Creating New Objects

3.7.1 Primary Shape Objects

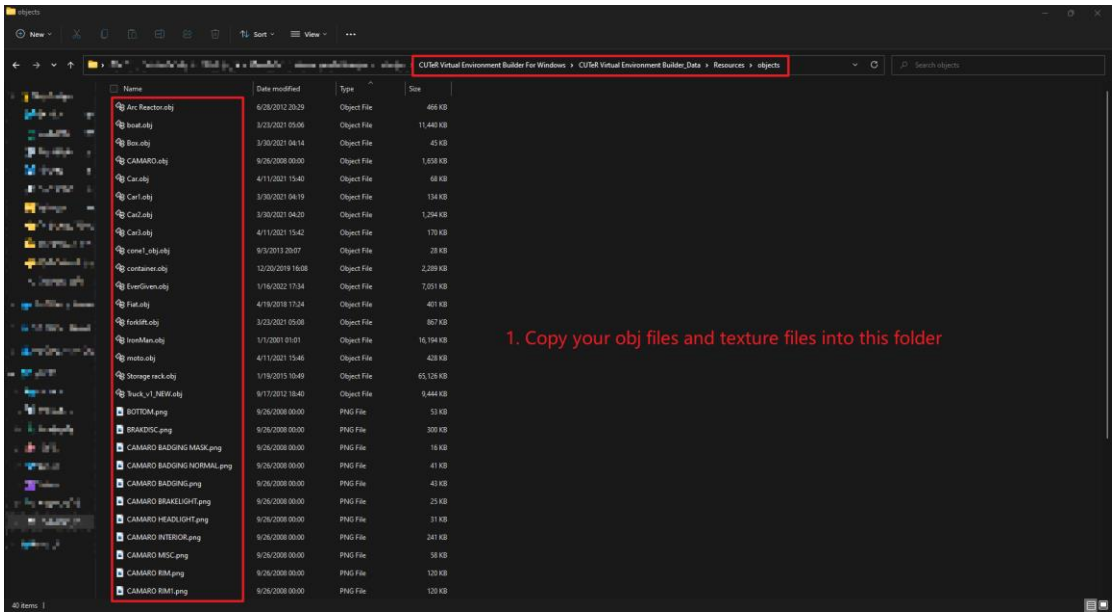
To create a primary shape object, open the ‘object menu’ and select one from the four primary options.

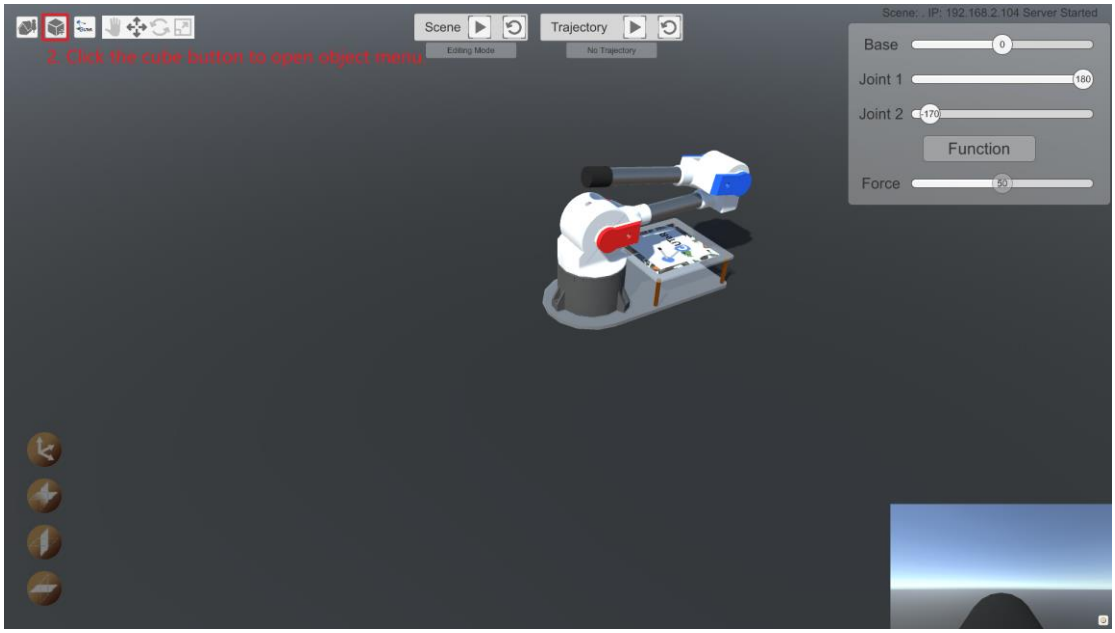


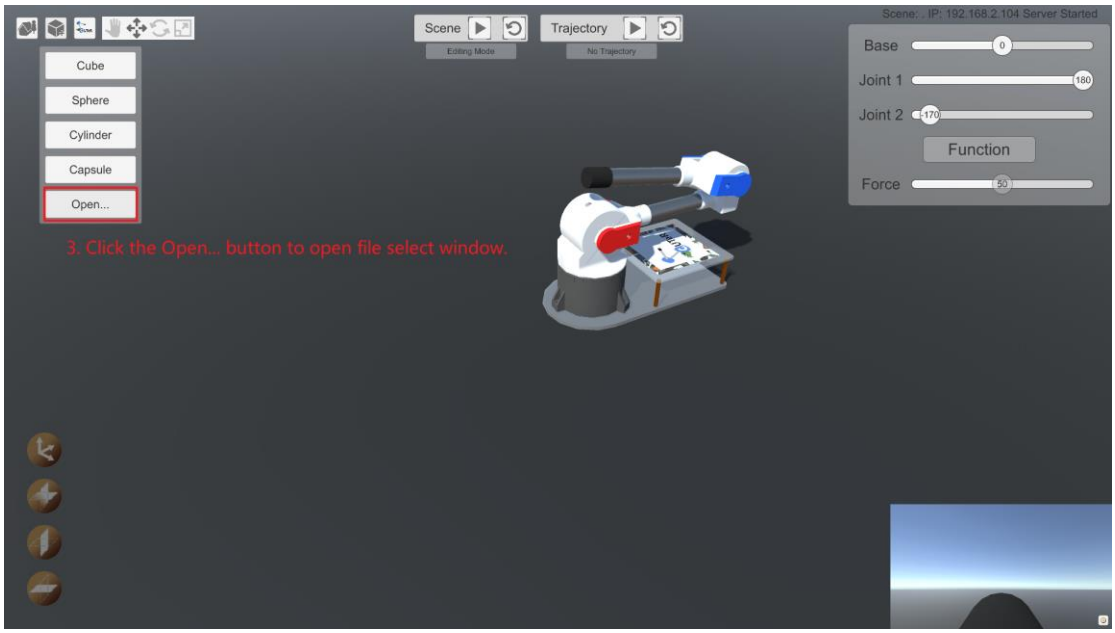


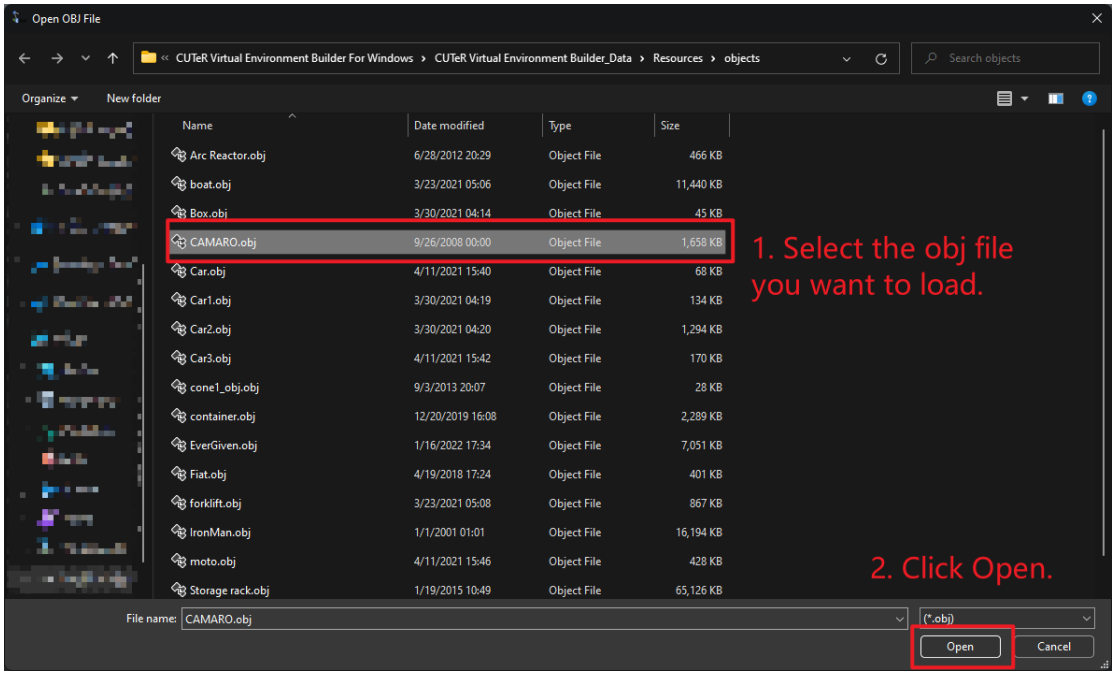
3.7.2 3D Models

To load an existing 3D obj file. First move your obj file and your texture files into the <application root folder>/CUTeR Virtual Environment Builder_Data/Resources/objects. Then select the 'Open...' option and then select your .obj file. The texture may not be loaded correctly, the object will be in white color if the texture loading fails.




2. Click the cube button to open object menu.
Scene
Editing Mode
Trajectory
No Trajectory
Scene: . IP: 192.168.2.104 Server Started
Base
0
Joint 1
180
Joint 2
-170
Function
Force
50

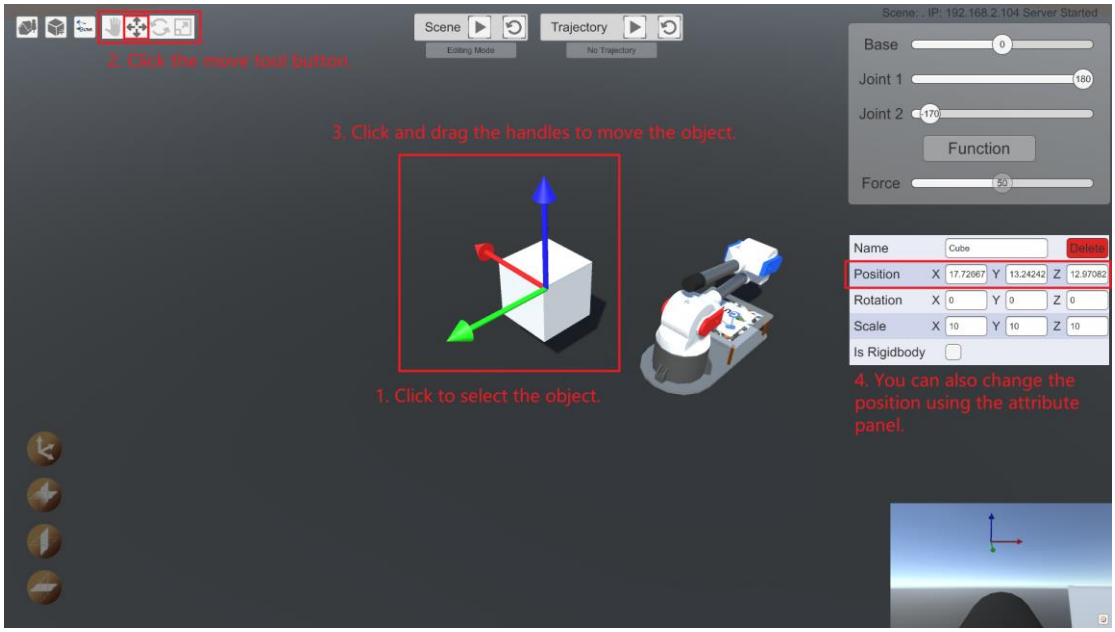

Cube
Sphere
Cylinder
Capsule
Open...
3. Click the Open... button to open file select window.
Scene
Editing Mode
Trajectory
No Trajectory
Scene: . IP: 192.168.2.104 Server Started
Base
0
Joint 1
180
Joint 2
-170
Function
Force
50



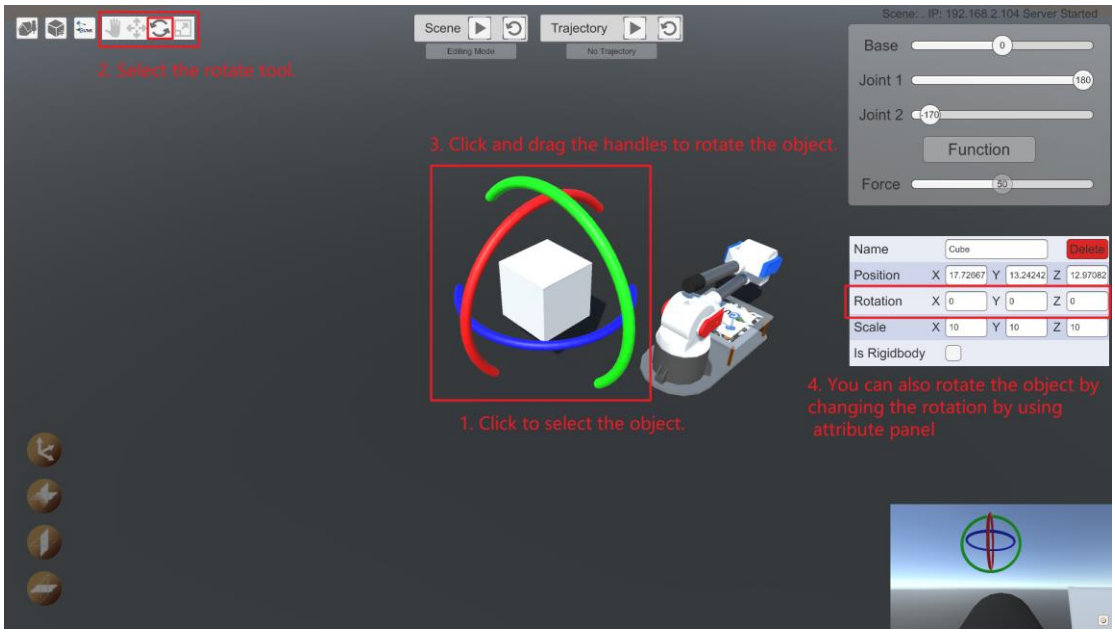
3.8 Editing Objects

You can move/rotate/scale the objects using different tools. You can also change some physical attributes of the object.

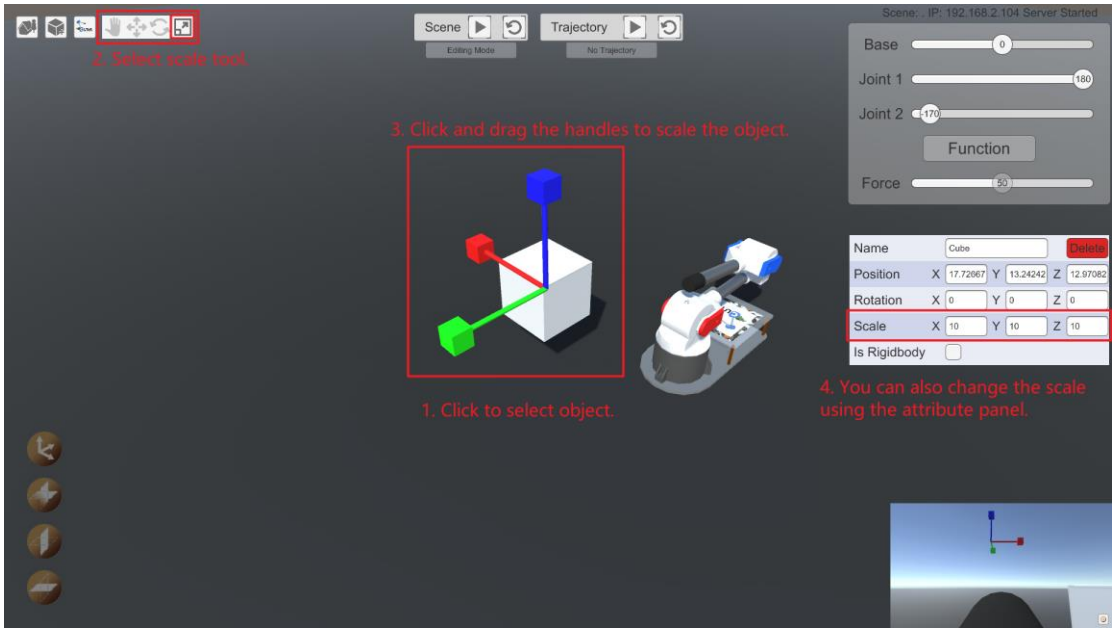
3.8.1 Move



3.8.2 Rotate



3.8.3 Scale



3.8.4 Physics

You can also use the attribute panel on the right-hand side of the screen to change the position/rotation/scale of the cube.

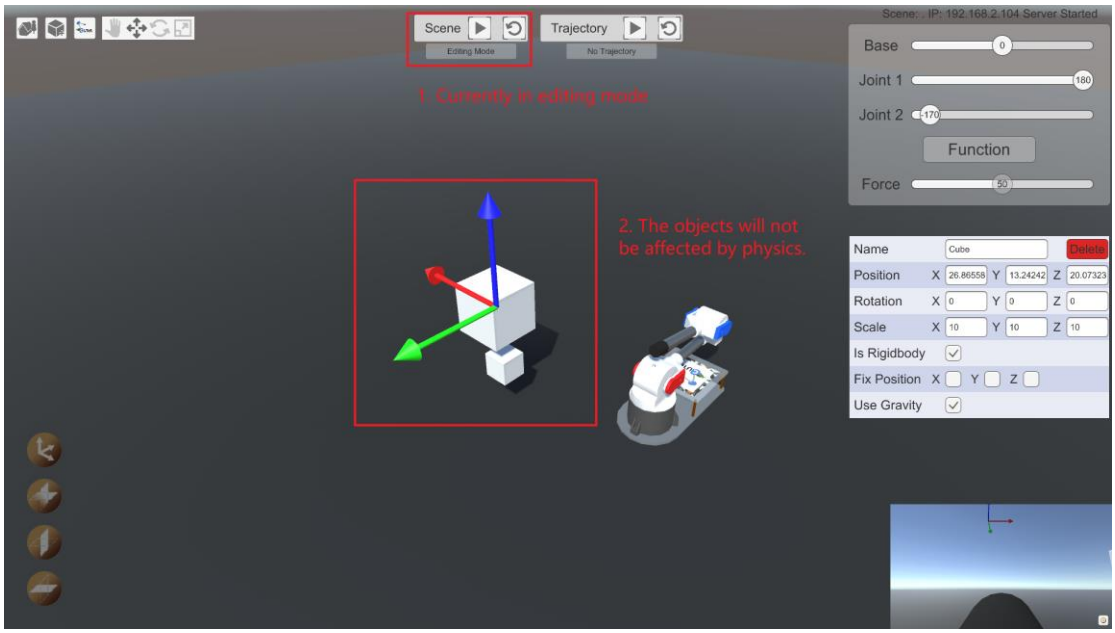
There are four extra tabs on the attribute panel, and their effects are

Name	Effect
Name	The name of the object. It should be unique, so later you can use it to generate dynamic trajectories.
Is Rigidbody	This toggle A Rigidbody object will be pulled downward by gravity (if selected) and will react to collisions with incoming objects if it is also a rigidbody object. An object without rigidbody has only visual effects.
Fix Position	Those toggles are for fixing the movement along axis.
Use Gravity	This toggle

3.9 Playing/Editing Mode

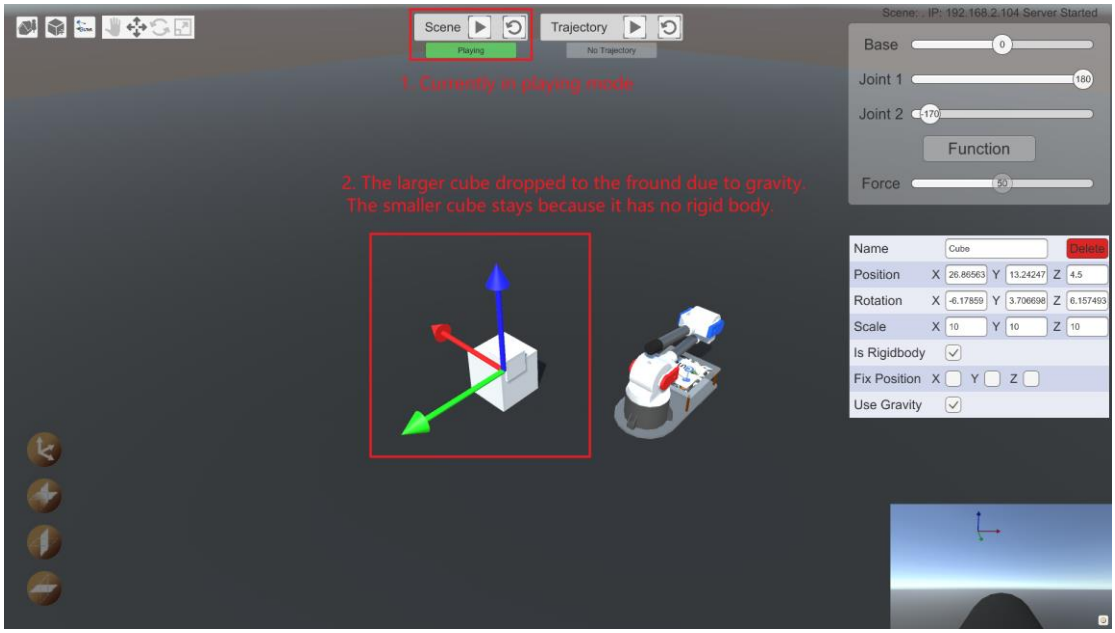
3.9.1 Editing Mode

This mode is for you to edit the virtual environment. In this mode, all the physical effects and all the end tools' function will be disabled.



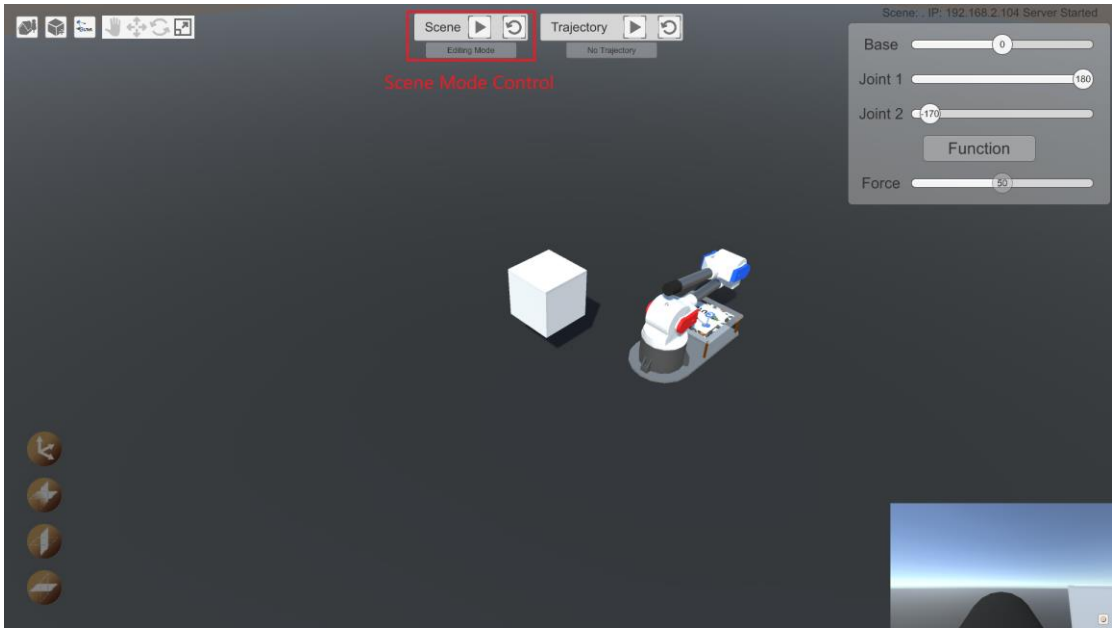
3.9.2 Playing Mode

This mode is for you to test the virtual environment. In this mode, all the physical effects will be enabled. And the end tools will act as normal.



3.9.3 Change Modes

You can change and check the current mode on the top of the screen. The left button is for entering the playing mode. The right button is for exiting the playing mode. The objects will restore after exiting the playing mode.

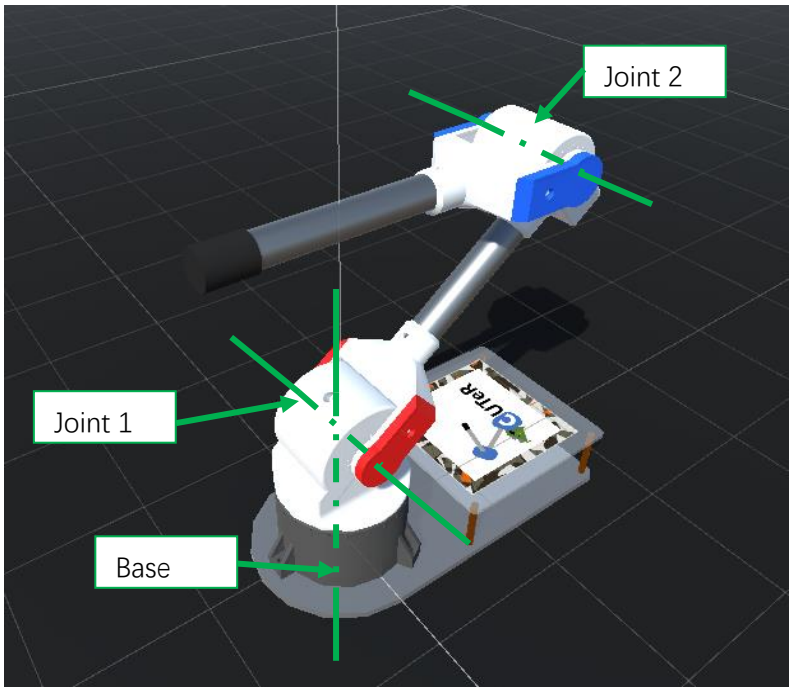


3.10 Knowing the robot

3.10.1 3DoF Robot

3.10.1.1 Joints

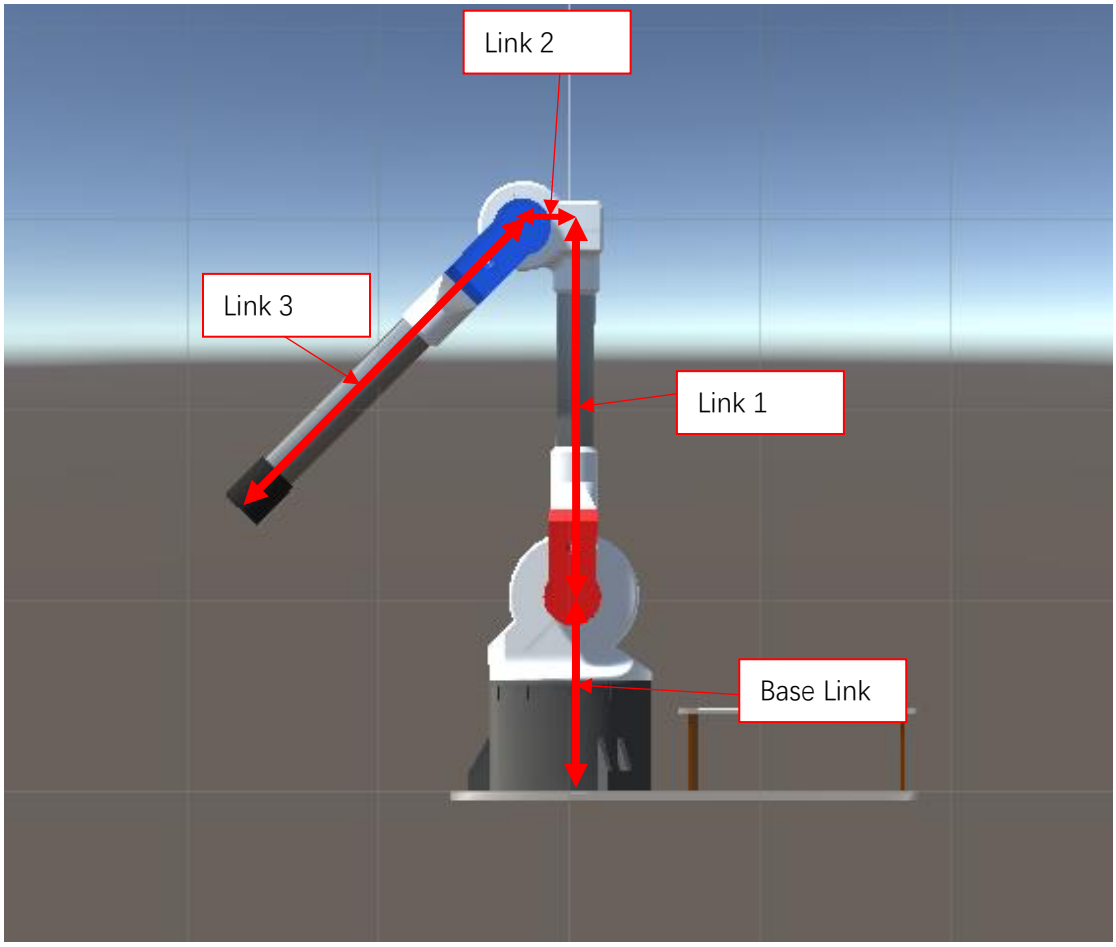
The robot has three joints, and their ranges are:



Name	Range (Degree)
Base Joint	-90 ~ 90
Joint 1	-15 ~ 180
Joint 2	-140 ~ 45

3.10.1.2 Links

The robot has four links, and their length are:

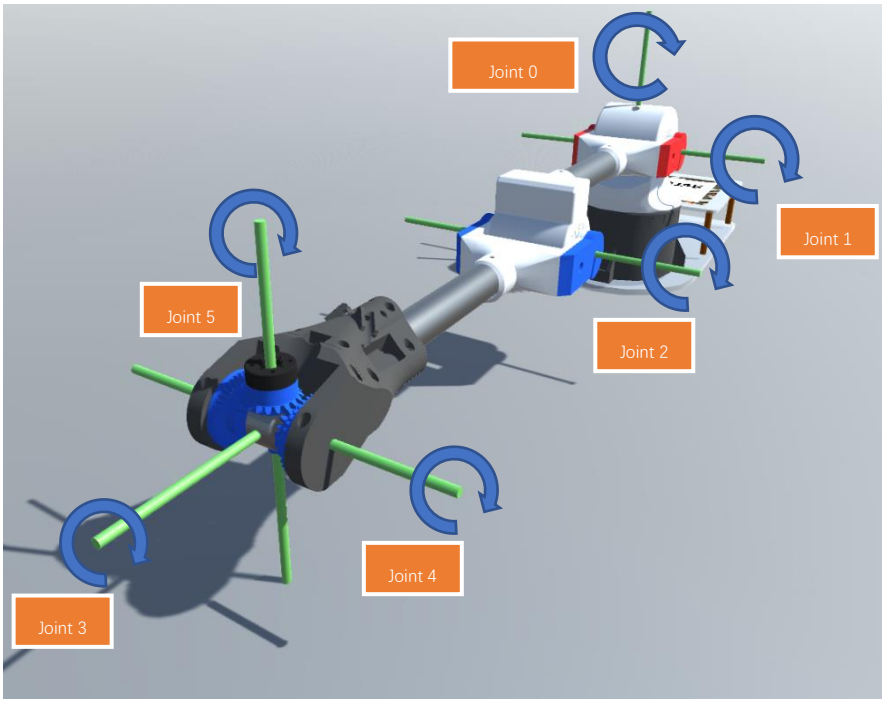


Name	Length (cm)
Base Link	10.18
Link 1	19.41
Link 2	2.91
Link 3	20.2

3.10.2 6DoF Robot

3.10.2.1 Joints

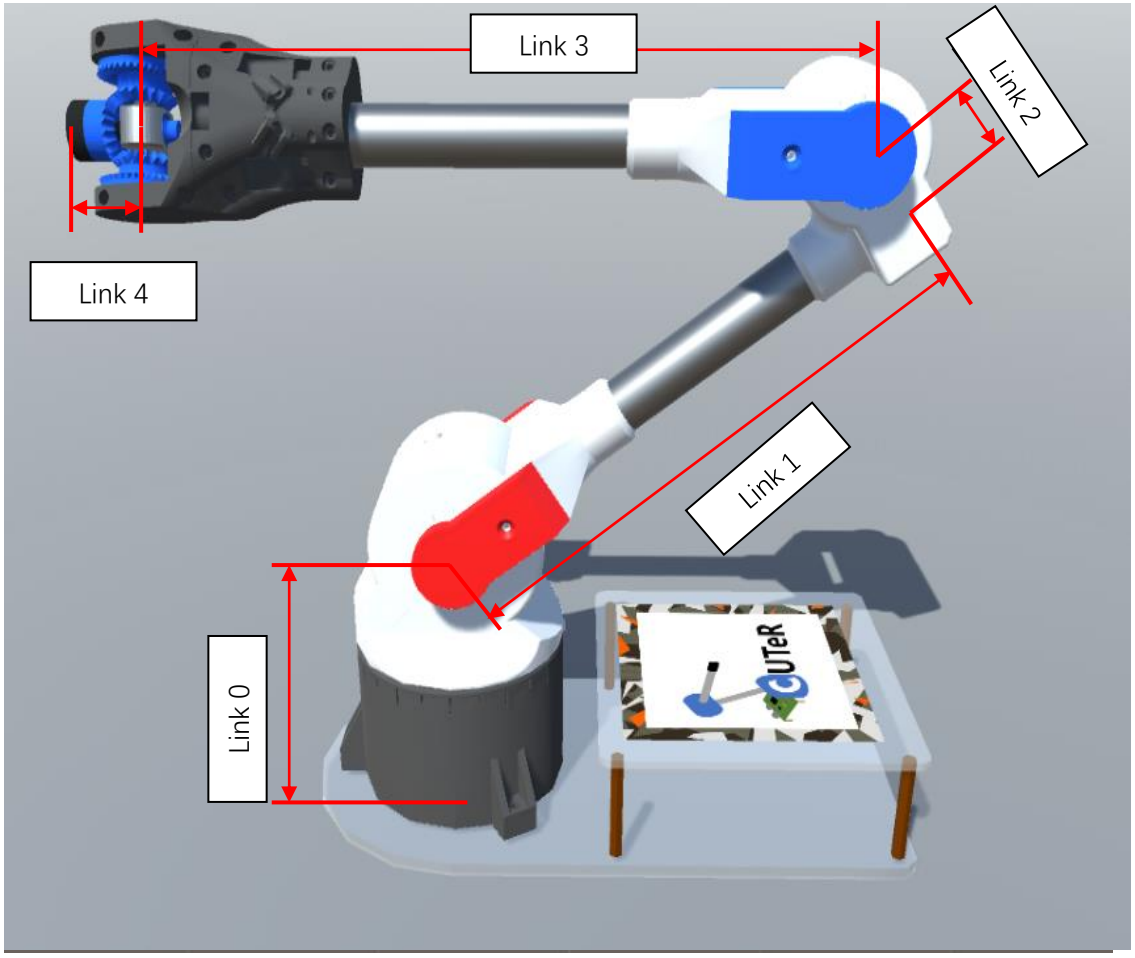
The robot has three joints, and their ranges are:



Name	Range (Degree)
Joint 0	-90 ~ 90
Joint 1	0 ~ 180
Joint 2	-140 ~ 45
Joint 3	-180 ~ 180
Joint 4	-110 ~ 110
Joint 5	-180 ~ 180

3.10.2.2 Links

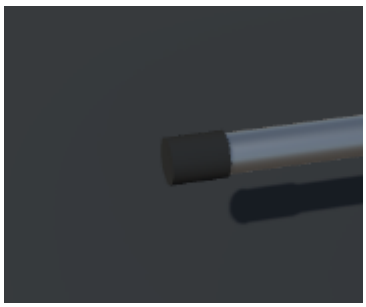
The robot has four links, and their length are:



Name	Length (cm)
Link 0	10.18
Link 1	19.41
Link 2	2.91
Link 3	25.22
Link 4	3.00

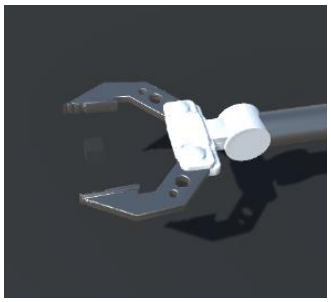
3.10.3 End Effectors

3.10.3.1 None

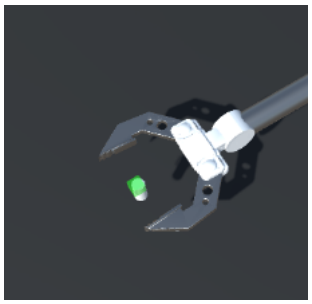


A black hat with no function.

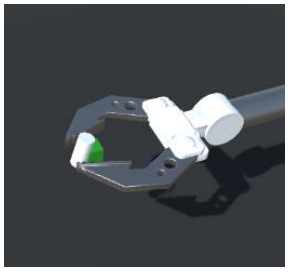
3.10.3.2 Grabber



Grabber at off state and no objects can be grabbed.



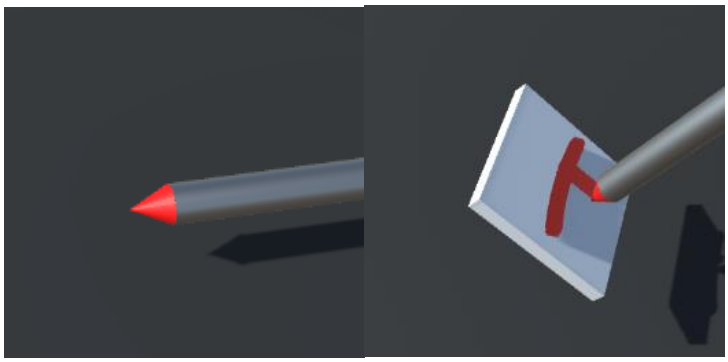
Grabber at off state and there are objects can be grabbed.



Grabber at on state with objects grabbed.

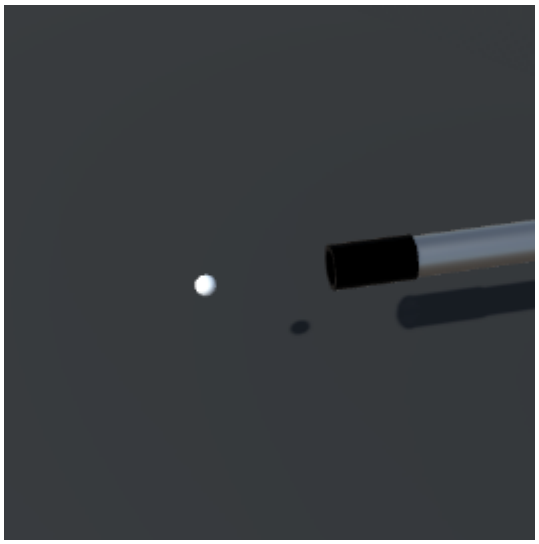
A robot grabber that can grab objects. There is a transparent indicator that will turn green when there is an object can be grabbed. (Only object with rigid body can be grabbed)

3.10.3.3 Pen



The pen can draw red lines on the objects with rigid body.

3.10.3.4 Launcher



The launcher can shoot a small white ball. You can change the initial force of the ball using the force slider in the joystick panel.

3.11 Operate the Robot Arm

3.11.1 Connect to Robot Arm Server (Optional)

To generate dynamic trajectories, you need to connect the robot arm server.
First you need to start the python server using PyCharm or other method.
Then open robot arm setting panel. Enter the server's IP and port.

Click connect to connect the robot server.

3.11.2 Control Robot Using Joystick

You can find the joystick on the top right corner of the screen.
There are five control widgets:

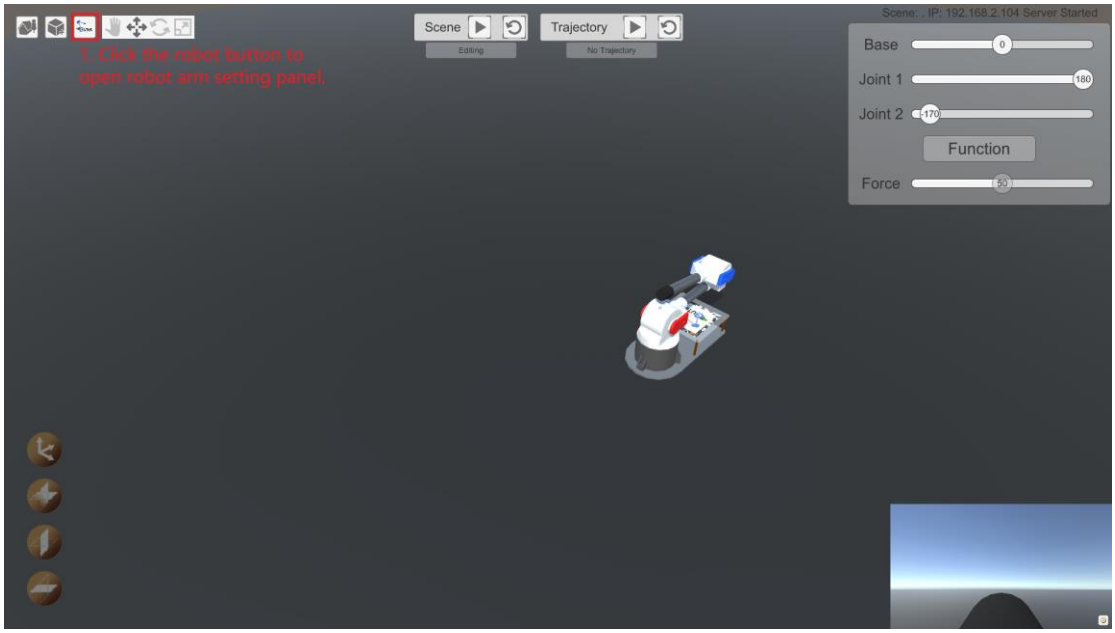
Name	Function
Base	Base joint angle slider. For controlling the base joint's angle.
Joint 1	Joint 1 angle slider. For controlling the joint1's angle.
Joint 2	Joint 2 angle slider. For controlling the joint2's angle.
Function	End tool function fire button. For grabber, it means grab/release. For launcher, it means fire.
Force	Launcher force slider. For controlling the launcher's fire force.

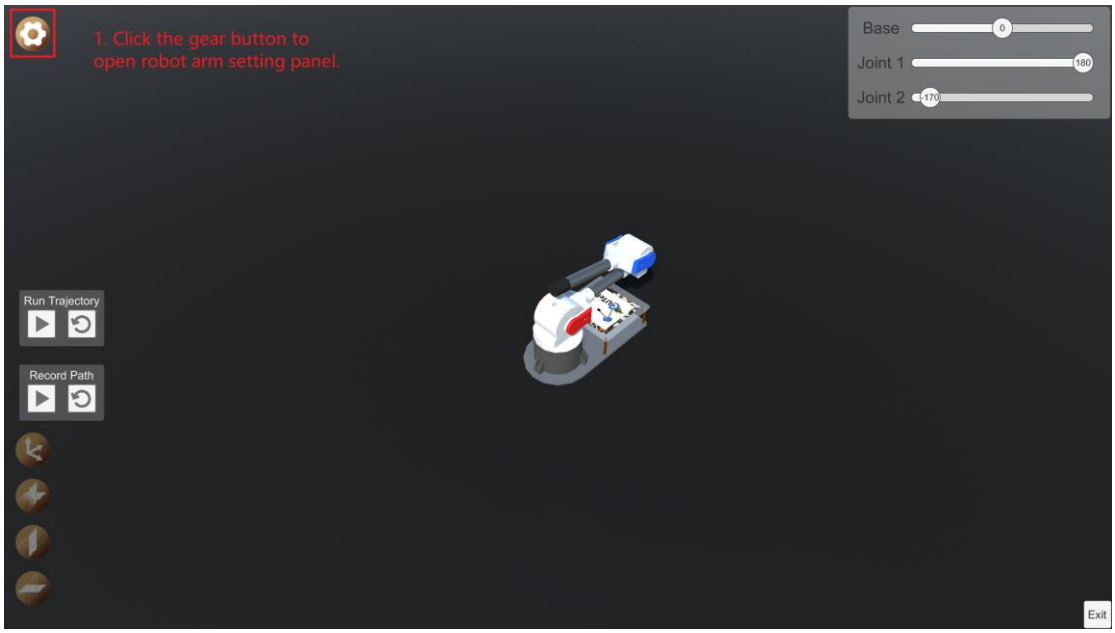
3.11.3 Control Robot Using Trajectory

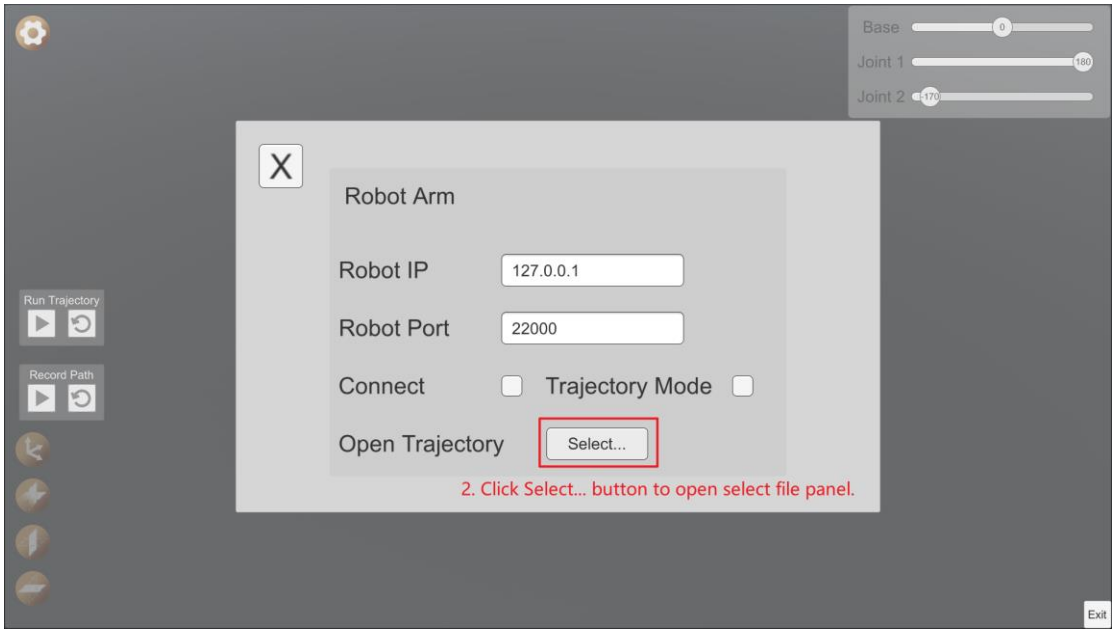
3.11.3.1 Load a static trajectory

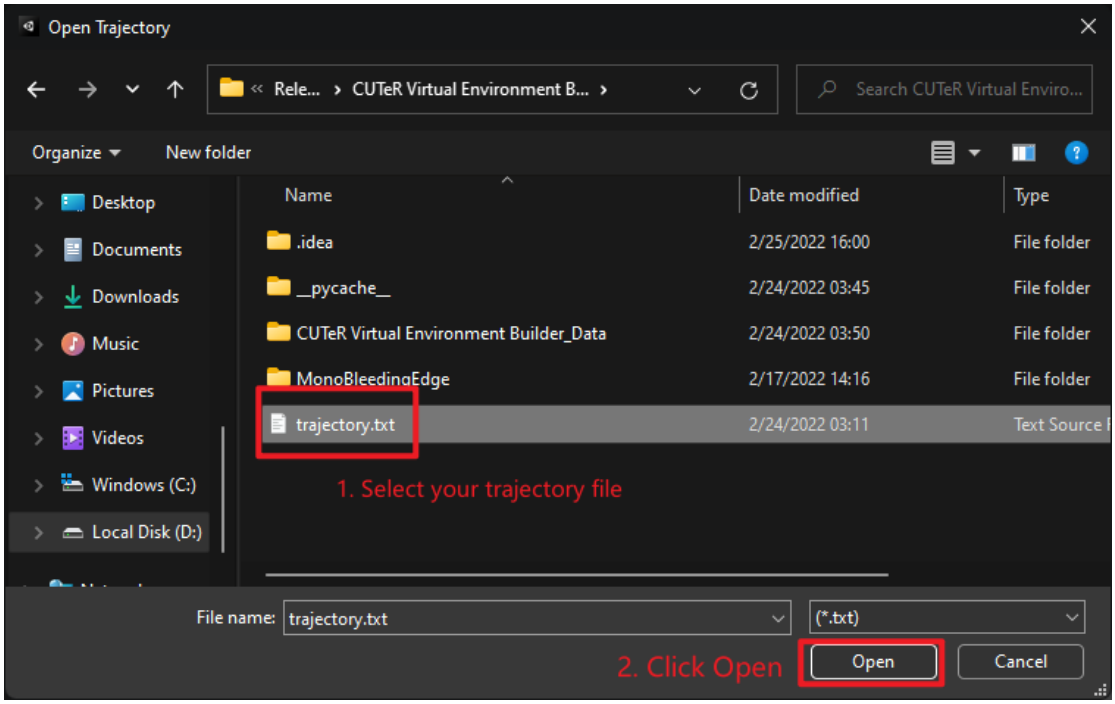
3.11.3.1.1 CUTeR Robot Arm Simulator for Windows / CUTeR Virtual Environment Builder / CUTeR Virtual Environment Simulator for Web

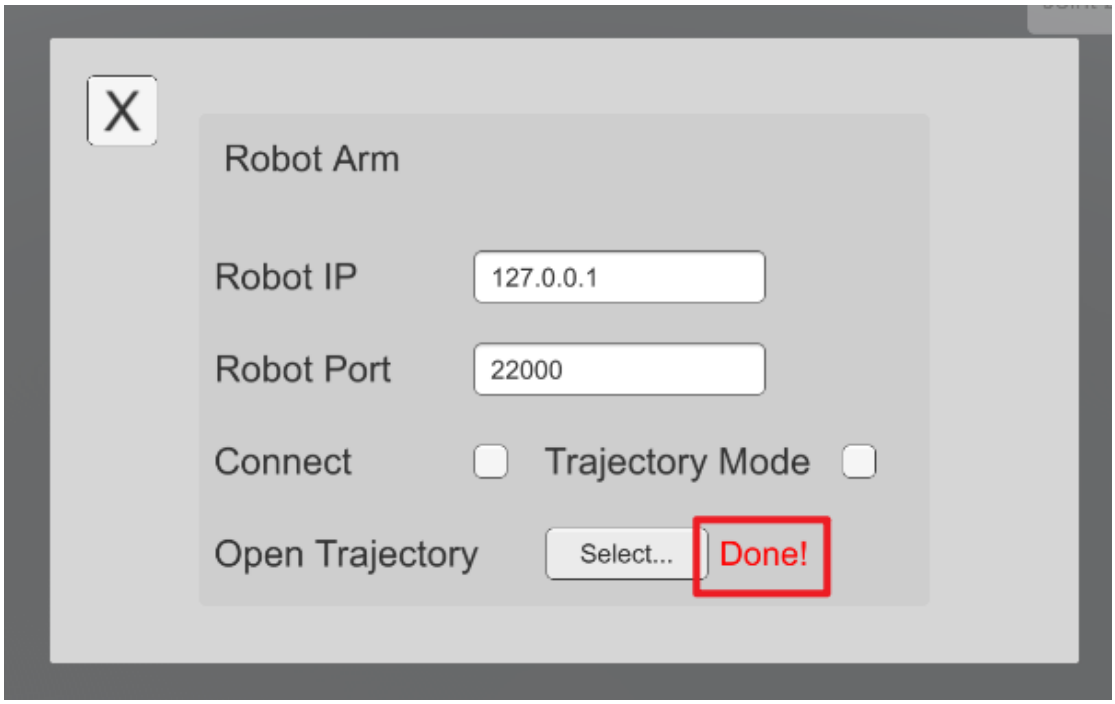
Click the CUTeR logo button or the gear button on the top left corner. The Robot Arm panel will pop up. Then click the Select button and then select your trajectory.txt file.


1. Click the robot button to
open robot arm setting panel.
Scene
Editing
Trajectory
No Trajectory
Scene: . IP: 192.168.2.104 Server Started
Base
0
Joint 1
180
Joint 2
-170
Function
Force
50


1. Click the gear button to
open robot arm setting panel.
Base
0
Joint 1
180
Joint 2
-170
Run Trajectory
Record Path
Exit

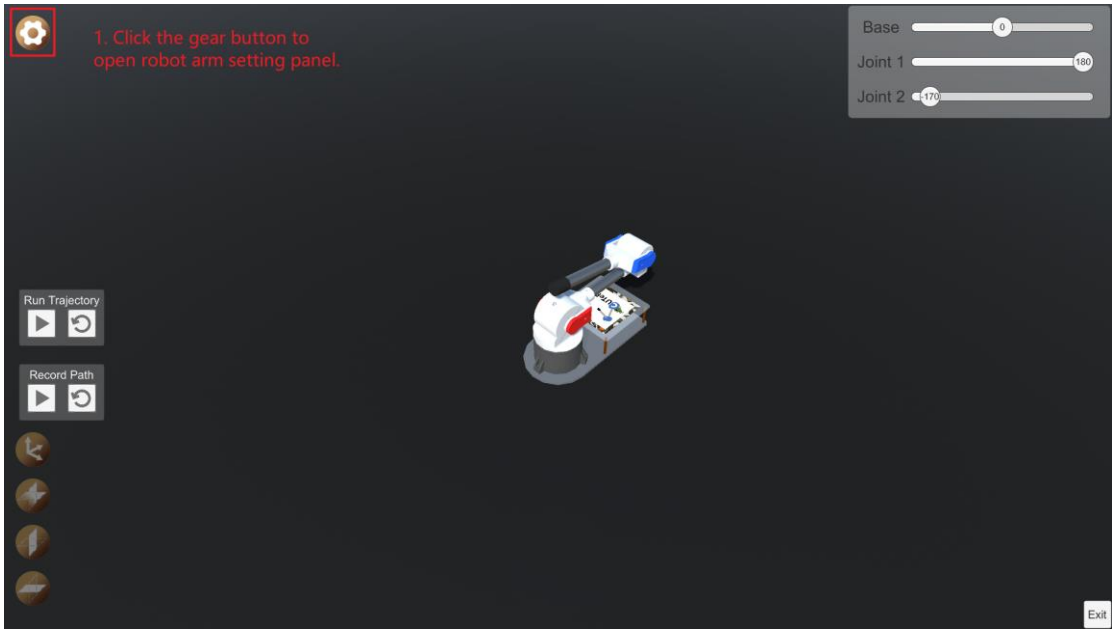

Base
Joint 1
Joint 2
Robot Arm
Robot IP
127.0.0.1
Robot Port
22000
Connect
Trajectory Mode
Open Trajectory
Select...
2. Click Select... button to open select file panel.
Run Trajectory
Record Path
Exit

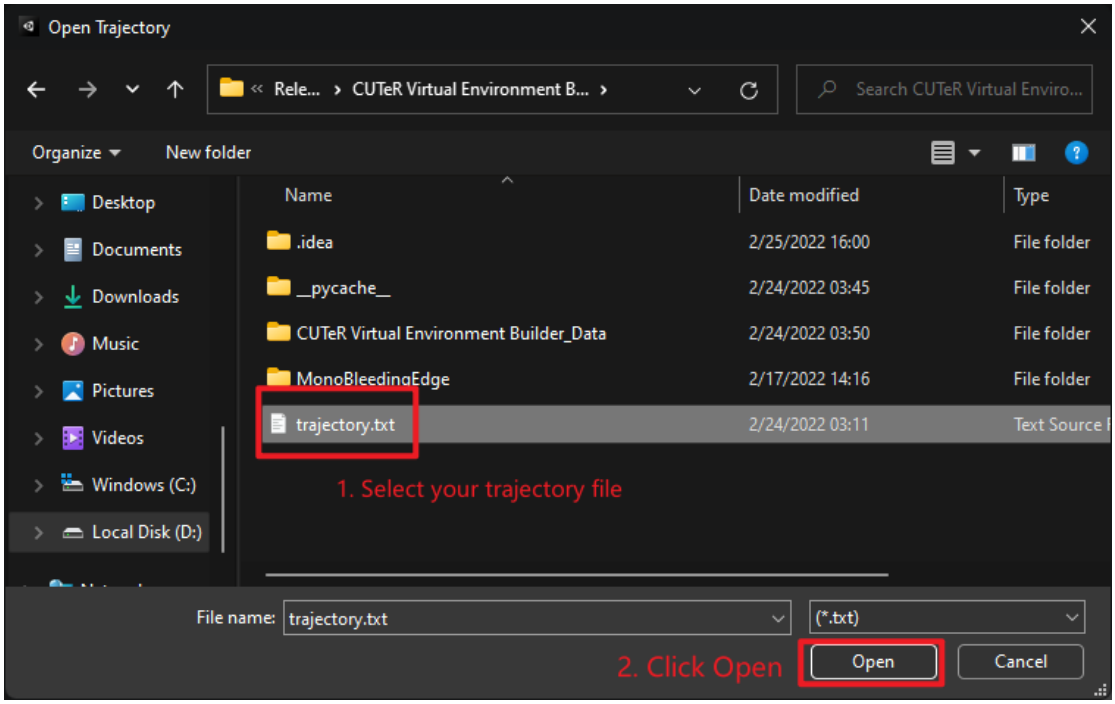

Open Trajectory
Rele... › CUTeR Virtual Environment B... ›
Search CUTeR Virtual Enviro...
Organize
New folder
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
Local Disk (D:)
Name
Date modified
Type
.idea
2/25/2022 16:00
File folder
__pycache__
2/24/2022 03:45
File folder
CUTeR Virtual Environment Builder_Data
2/24/2022 03:50
File folder
MonoBleedingEdge
2/17/2022 14:16
File folder
trajectory.txt
2/24/2022 03:11
Text Source
1. Select your trajectory file
File name:
trajectory.txt
(*.txt)
2. Click Open
Open
Cancel



3.11.3.1.2 CUTeR Arm Web Simulator

Click the CUTeR logo button or the gear button on the top left corner. Then select your trajectory.txt file.





3.11.3.2 Load a dynamic trajectory

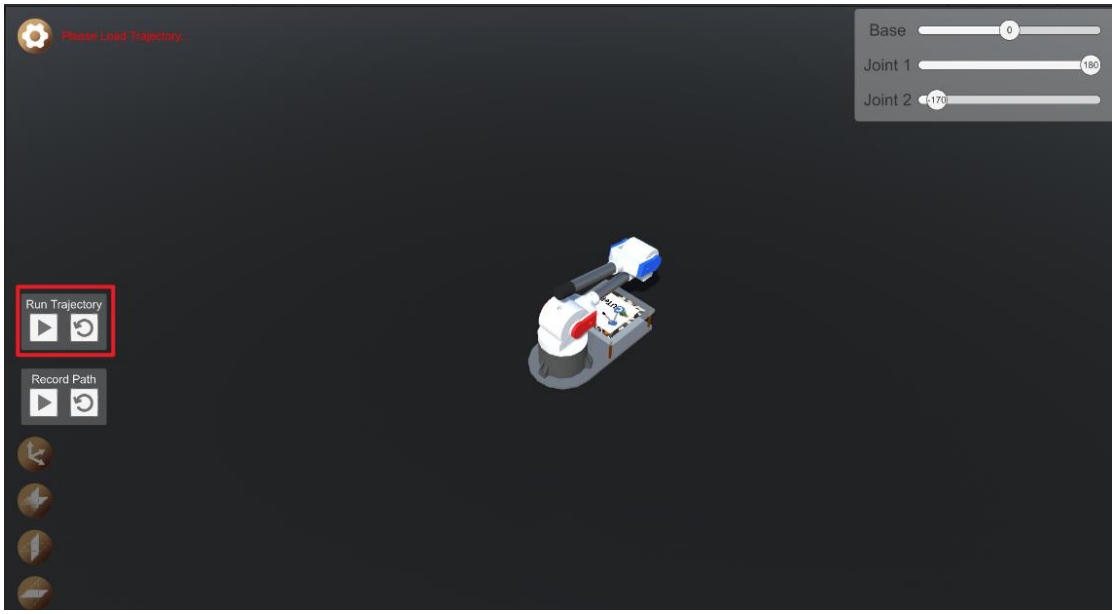
At this point, the dynamic trajectory is not really real-time generated.

1. You need to connect to the robot arm server (python) and turn the trajectory mode to use dynamic trajectory. You can turn on the trajectory mode in the robot arm setting panel.
2. Click the play button in the trajectory control panel. The robot will follow the trajectory generated in Trajectory.py.
3. You can load pre-defined static trajectories or real-time generated dynamic trajectories at the same time but only one trajectory will be executed. If you turn on the Trajectory Mode, the simulator will use the dynamic trajectory, otherwise it will use the static trajectory.

3.11.3.3 Run the trajectory

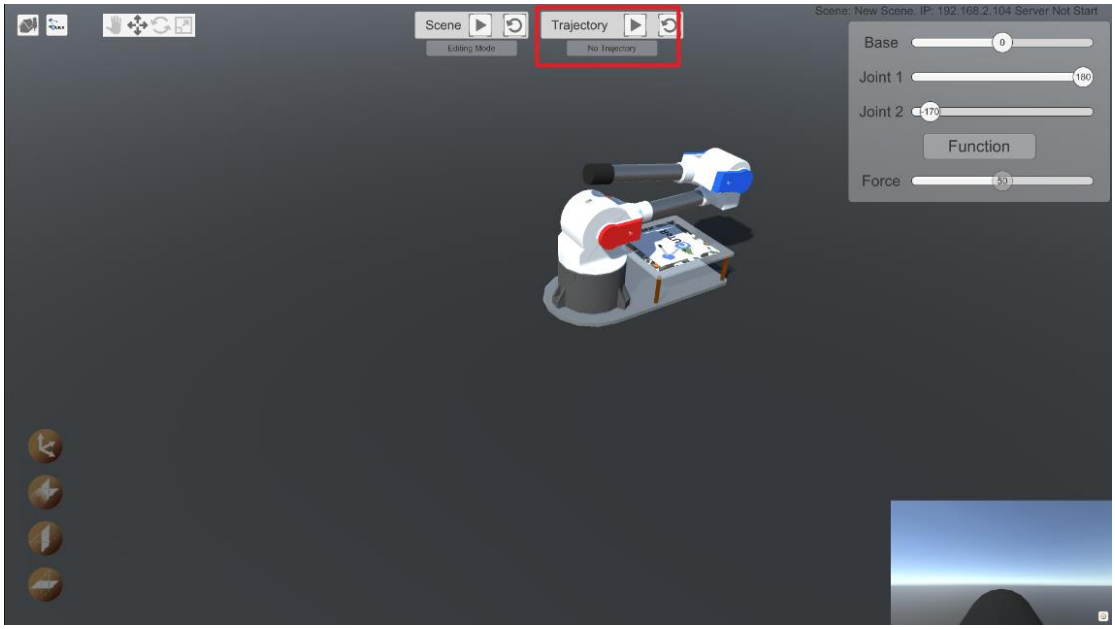
3.11.3.3.1 CUTeR Arm Robot Simulator

You can find the trajectory control panel on the right side of the screen. The first button is the play/pause button. The second button is the reset button. Click the play button to run the trajectory.



3.11.3.3.2 Virtual Environment Builder

You can find the trajectory control panel on the top of the screen. The first button is the play/pause button. The second button is the reset button. Click the play button to run the trajectory.



4 Generate Trajectory

You will use python to generate the trajectories.
In the file Trajectory.py. You can find one variable and two functions.

4.1 Function generate_trajectory

This function is for generate trajectories that using joint angles.

The return value of this function is a 2-dimensional list. The first dimensional represents the three joints of the robot. The second dimension represents the frames of each joint. Here is a sample of the result:

(base angle from 4 to- 5, angle 1 from 0 to 9, angle 2 from -10 to -1)

```
[ [4, 3, 2, 1, 0, -1, -2, -3, -4, -5]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1] ]
```

To fire the functional tool, add “fire” to every joint. You can do it like this:

```
angle_0_list.append("fire")
angle_1_list.append("fire")
angle_2_list.append("fire")
```

And here is a sample of the result:

```
[ [4, 3, 2, 1, 0, -1, -2, -3, -4, -5, 'fire']
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'fire']
[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 'fire'] ]
```

4.2 Function generate_trajectory_cartesian_point

This function is for generate trajectories that using cartesian points, the simulator will calculate the IK for you, you can use this to check correctness of your IK. The return value of this function is a 2-dimensional list. The first dimensional represents the three axes. The second dimension represents the point of each axis for each frame.

Here is a sample of the result:

(x from 0.1 to 0.9, y stays at 0, y from 0.1 to 0.9)

```
[ [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
[0, 0, 0, 0, 0, 0, 0, 0, 0]
[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] ]
```

To fire the functional tool, add “fire” to every axis. You can do it like this:

```
x_list.append("fire")
y_list.append("fire")
z_list.append("fire")
```

And here is a sample of the result:

```
[ [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 'fire']
[0, 0, 0, 0, 0, 0, 0, 0, 0, 'fire']
[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 'fire'] ]
```

4.3 Variable using_cartesian_point

This variable is the switch of using joint angle or the cartesian point.
If the value is true, the cartesian point trajectory will be generated.
If the value is false, the joint angle trajectory will be generated.