1. Software Installation & Initial Setup

2.1 Installing CoppeliaSim

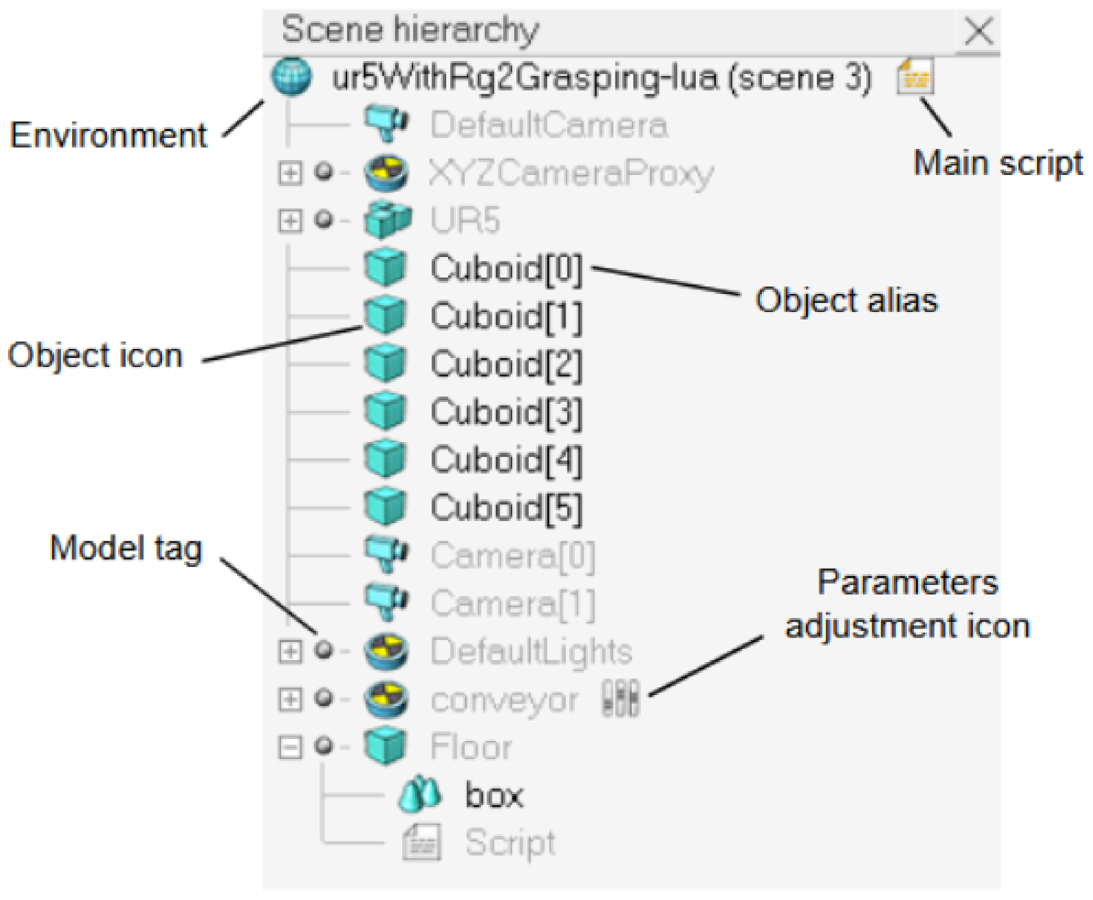
- Follow the instructions provided in the installation guide
- This guide walks you through **setup** → **simulation** → **scripting** → **competition rules**.

2.2 Basic Navigation in CoppeliaSim

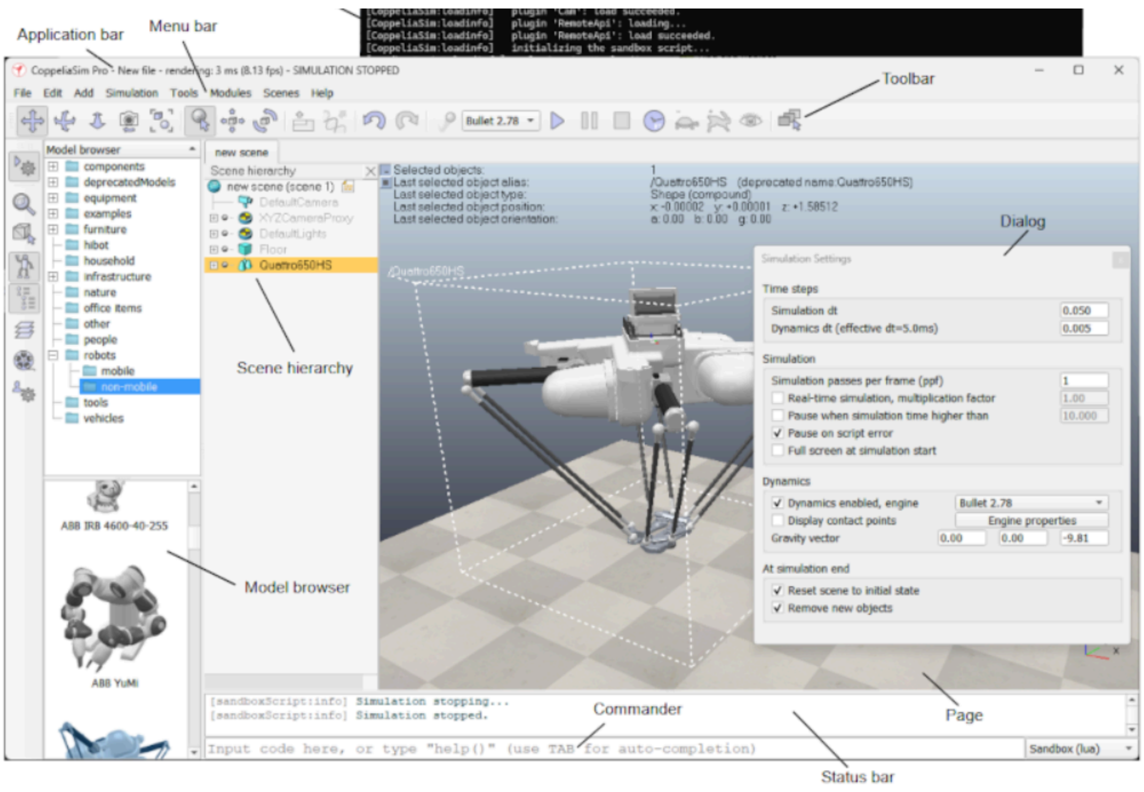
After launching CoppeliaSim:

Main UI Areas

- **Scene Hierarchy** (left): Objects, joints, scripts



- **3D Viewport** (center): Simulation world



CoppeliaSim Pro - New file - rendering: 3 ms (8.13 fps) - SIMULATION STOPPED

```
[CoppeliaSim:loadinfo]   plugin 'Cam': load succeeded.
[CoppeliaSim:loadinfo]   plugin 'RemoteApi': loading...
[CoppeliaSim:loadinfo]   plugin 'RemoteApi': load succeeded.
[CoppeliaSim:loadinfo]   initializing the sandbox script...
```

```
Selected objects:                    1
Last selected object alias:          /Quattro650HS   (deprecated name:Quattro650HS)
Last selected object type:           Shape (compound)
Last selected object position:       x: -0.00002   y: +0.00001   z: +1.58512
Last selected object orientation:    a: 0.00   b: 0.00   g: 0.00
```

Simulation Settings

Time steps
- Simulation dt: 0.050
- Dynamics dt (effective dt=5.0ms): 0.005

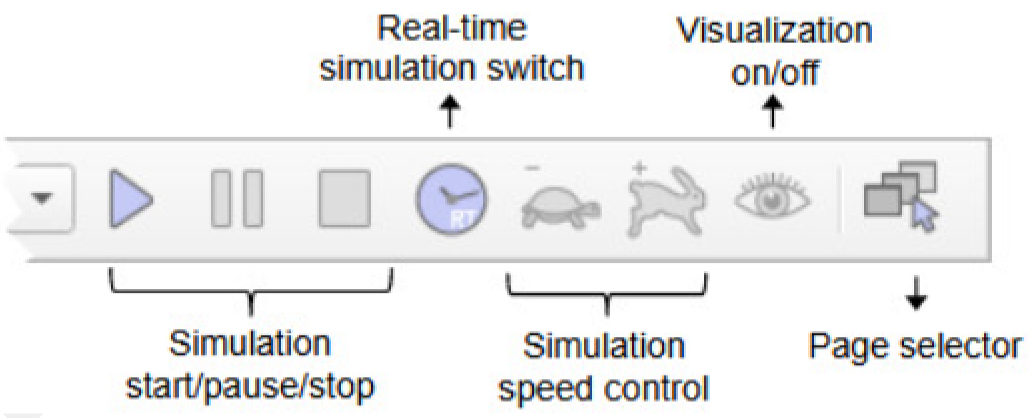
Simulation
- Simulation passes per frame (ppf): 1
- ☐ Real-time simulation, multiplication factor: 1.00
- ☐ Pause when simulation time higher than: 10.000
- ✓ Pause on script error
- ☐ Full screen at simulation start

Dynamics
- ✓ Dynamics enabled, engine: Bullet 2.78
- ☐ Display contact points — Engine properties
- Gravity vector: 0.00 0.00 -9.81

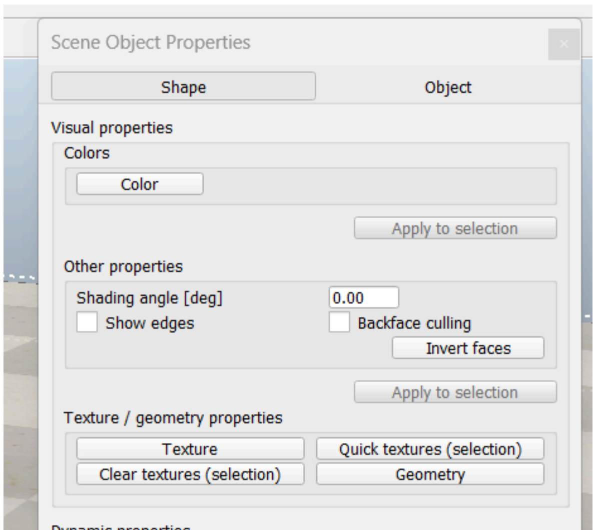
At simulation end
- ✓ Reset scene to initial state
- ✓ Remove new objects

```
[sandboxScript:info] Simulation stopping...
[sandboxScript:info] Simulation stopped.
Input code here, or type "help()" (use TAB for auto-completion)    Sandbox (lua)
```

- **Toolbar** (top): Run / Stop / Step simulation



- **Properties Panel** (right): Object & script properties



Scene Object Properties

Shape | Object

Visual properties

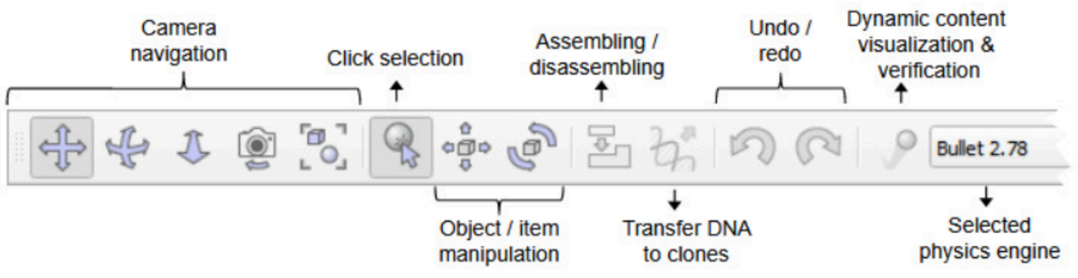
Colors
- Color
- Apply to selection

Other properties
- Shading angle [deg]: 0.00
- ☐ Show edges
- ☐ Backface culling
- Invert faces
- Apply to selection

Texture / geometry properties
- Texture
- Quick textures (selection)
- Clear textures (selection)
- Geometry

Dynamic properties

Toolbar view controls



- Shift positions of the selected object
- Rotate the orientation of the selected object
- Pan the camera navigation
- Rotate the camera view

2.3 Opening the Provided Scene

1. Go to **File** → **Open Scene**
2. Load the provided ROBOSOCCER scene (`.ttt`)
3. You will see:
    - Soccer field
    - Ball at centre
    - Robot (bot)
    - Goal post

---

3. Understanding Objects & Models

3.1 Objects in the Scene

Each entity in the scene is an **object**:

- Robot body
- Wheels
- Ball

- Goal post

Each object has:

- A **handle** (unique ID)
- Position & orientation
- Optional **child scripts**

3.2 Models

The robot is a **model**, which is a group of objects bundled together.

> You will **attach your script to the robot model**, not the entire scene.

---

4. Simulation Scripts – Basics

4.1 Script Objects

- **Simulation scripts:** A simulation script represents code that handles a particular function in a simulation. It runs only when the simulation is running.
- **Customisation scripts:** A customisation script represents code that handles a particular customisation aspect in a scene or model. It also runs when the simulation is not running.

4.2 Script Types

- Simulation Script **(Child Script running in simulation)**
- Add-on Script

CoppeliaSim uses **callback-based simulation scripts** that run automatically at specific phases of the simulation loop.

- `sysCall_init()` runs **once at the start** and is used for initialization (getting object handles, setting parameters).

- `sysCall_actuation()` runs **every simulation step** during the actuation phase and is meant for **applying control actions** (e.g., motor velocities).

- `sysCall_sensing()` also runs every step but during the sensing phase, used only for **reading sensors or simulation data**.

- `sysCall_thread()` enables **threaded execution** for sequential logic with delays, while

- `sysCall_cleanup()` runs **once at the end** for cleanup tasks when the simulation stops.

Scripts are edited using the **built-in Script Editor**, opened by double-clicking a script object in the scene hierarchy. The editor supports auto-completion, syntax highlighting, API tooltips, search/replace, and automatic saving—no manual save is required. Scripts can also include **external Lua or Python files** using `include` or `require`, allowing modular code reuse.

4.2 Supported Script Languages

CoppeliaSim supports:

- **Lua** (native, default)
- Python

For ROBOSOCCER 4.0:

- Robot logic → **Lua/Python child script**
- Browser control → **JavaScript via WebSocket API**

---

5. Python Configuration (Mandatory Setup)

To enable Python scripting:

You must install Python on your system. Recommended Python version 3.10.XX

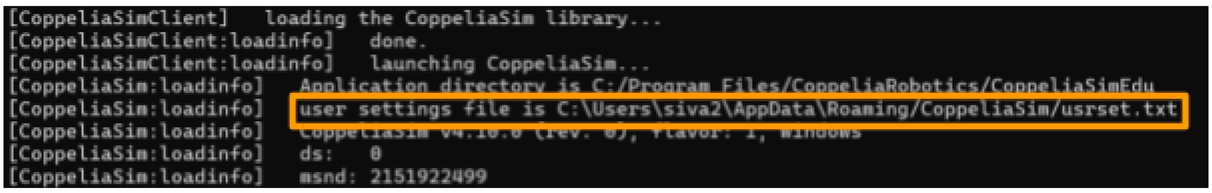
Download Link: Python 3.10.10

Once downloaded, verify the installation using the command.

Open Terminal and run

```
python --version
```

Find the `usrset.txt` file location. When you start Coppeliasim, a terminal window opens along with it.

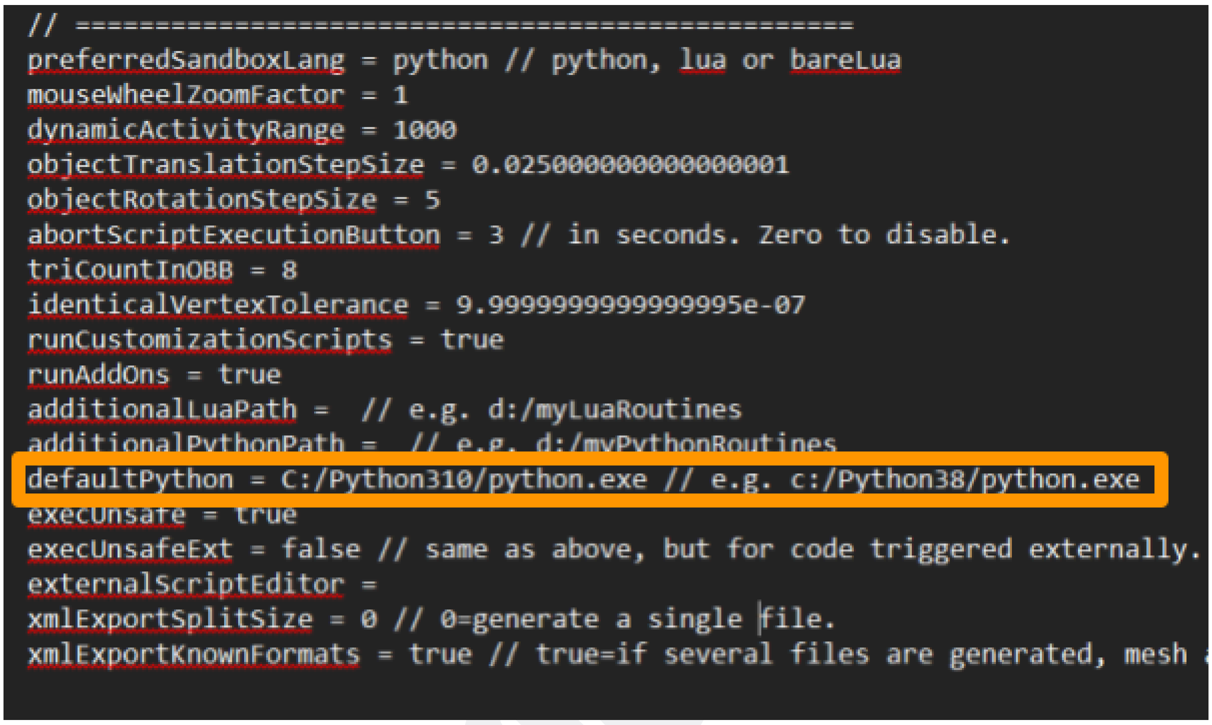


```
[CoppeliaSimClient]   loading the CoppeliaSim library...
[CoppeliaSimClient:loadinfo]   done.
[CoppeliaSimClient:loadinfo]   launching CoppeliaSim...
[CoppeliaSim:loadinfo]   Application directory is C:/Program Files/CoppeliaRobotics/CoppeliaSimEdu
[CoppeliaSim:loadinfo]   user settings file is C:\Users\siva2\AppData\Roaming/CoppeliaSim/usrset.txt
[CoppeliaSim:loadinfo]   CoppeliaSim v4.10.0 (rev. 0), flavor: 1, Windows
[CoppeliaSim:loadinfo]   ds:   0
[CoppeliaSim:loadinfo]   msnd: 2151922499
```

Note the location of file from the terminal windows

5.1 Configure Python Interpreter

1. Open `usrset.txt`
2. Locate the Python section



```
// ==================================================
preferredSandboxLang = python // python, lua or bareLua
mouseWheelZoomFactor = 1
dynamicActivityRange = 1000
objectTranslationStepSize = 0.025000000000000001
objectRotationStepSize = 5
abortScriptExecutionButton = 3 // in seconds. Zero to disable.
triCountInOBB = 8
identicalVertexTolerance = 9.9999999999999995e-07
runCustomizationScripts = true
runAddOns = true
additionalLuaPath =  // e.g. d:/myLuaRoutines
additionalPythonPath =  // e.g. d:/myPythonRoutines
defaultPython = C:/Python310/python.exe // e.g. c:/Python38/python.exe
execUnsafe = true
execUnsafeExt = false // same as above, but for code triggered externally.
externalScriptEditor =
xmlExportSplitSize = 0 // 0=generate a single file.
xmlExportKnownFormats = true // true=if several files are generated, mesh
```

Replace the python execution file with your installed python .exe path

3. Set Python 3.10 path:

pythonExecutable = C:/Python310/python.exe

4. Save and restart CoppeliaSim

✔ Python 3.10 is **mandatory** for compatibility.

6. Connectivity & Communication Modules

Open:
Menu → Modules → Connectivity

Enabled Modules

6.1 Visual Web Stream API

- Purpose: Live video streaming to the browser
- Default Port: **23020**
- Used for real-time simulation view

6.2 WebSocket Remote API

- Purpose: Control simulation from the browser
- Default Port: **23050**
- Used to send movement commands

---

7. Robot Control – Core Task

7.1 Child Script Attachment

- Expand the robot model in the Scene Hierarchy
- Right-click → Add → Child Script → Non-Threaded

7.2 Wheel Control Concept

The robot moves by controlling **wheel joints**.

Typical steps:

1. Get wheel joint handles
2. Set target velocity
3. Change velocity based on commands

Example actions:

- Forward
- Backward
- Turn left
- Turn right

---

8. Browser-Based Control UI

8.1 Features

- Live simulation video stream
- Keyboard-based robot control

8.2 Key Mapping

Key	Action
W	Forward
S	Backward
A	Turn Left
D	Turn Right

The browser:

- Sends commands via **WebSocket**
- Receives no physics data (control only)

Note: ROBOSOCCER 4.0 Organising team will provide the controller UI through a GitHub repo. Students do not need to build the controller UI. They can clone the repo and run the webpage on their local host.

We highly encourage all the participants to explore the manuals from the CoppeliaSim Official website. It is a great learning resource to familiarise yourself with the software.

Prepared by

ROBOSOCCER 4.0

TEAM

In case you find any errors or misinterpreted data presented, kindly let us know
23f3100008@es.study.iitm.ac.in *23f3100008@es.study.iitm.ac.in*