

Omniverse Base Workshop 1

KUKA X IPI

- 2x 4h Course with examples to
 - Learn the Basics in Omniverse, USD & Isaac Sim
 - Enable you to do Robotic model preparation & Simulation
 - Introduction to IsaacLab



Trainer Introduction



Fabian Fichtl



Julian Zürn



Institut für Produktion und Informatik

Technologietransferzentrum der Hochschule Kempten

- anwendungsorientierte Forschungseinrichtung
- F&E-Projekte im Kontext Digitalisierung in der Produktion
- Seit 2021 mit 30 Mitarbeitern
- Technologietransfer und Startup Inkubator
- VIBN und Industrial Metaverse in der Lehre





Introduction round

Please tell a few sentences about yourself,
what you expect from this workshop and
what you want to use Omniverse for

Agenda Tag 1:

- Introduction Metaverse
- Omniverse basics:
Nucleus | Data Formats | PhysX | extensions | Stage
- 5 min Pause
- Hands on: Scene manipulation, Kinematics and Assets
 - UI Manipulation Navigation, Manipulation (property window)
 - Mass & Mesh manipulation
 - First Robot
- 10 min Pause
- Import and Tune a Robot
- 5 min Pause
- First Robot Control (LULA test widget)
 - Adding Sensors and Kameras

Agenda Tag 2:

- Debugging Tools:
 - Console, VS Code, UI
- Scene Architecture
 - Build your Scenario: Robot, Gripper, Controller, Environment, Handle Object
- Pick & Place
- Information Channels | Further Courses



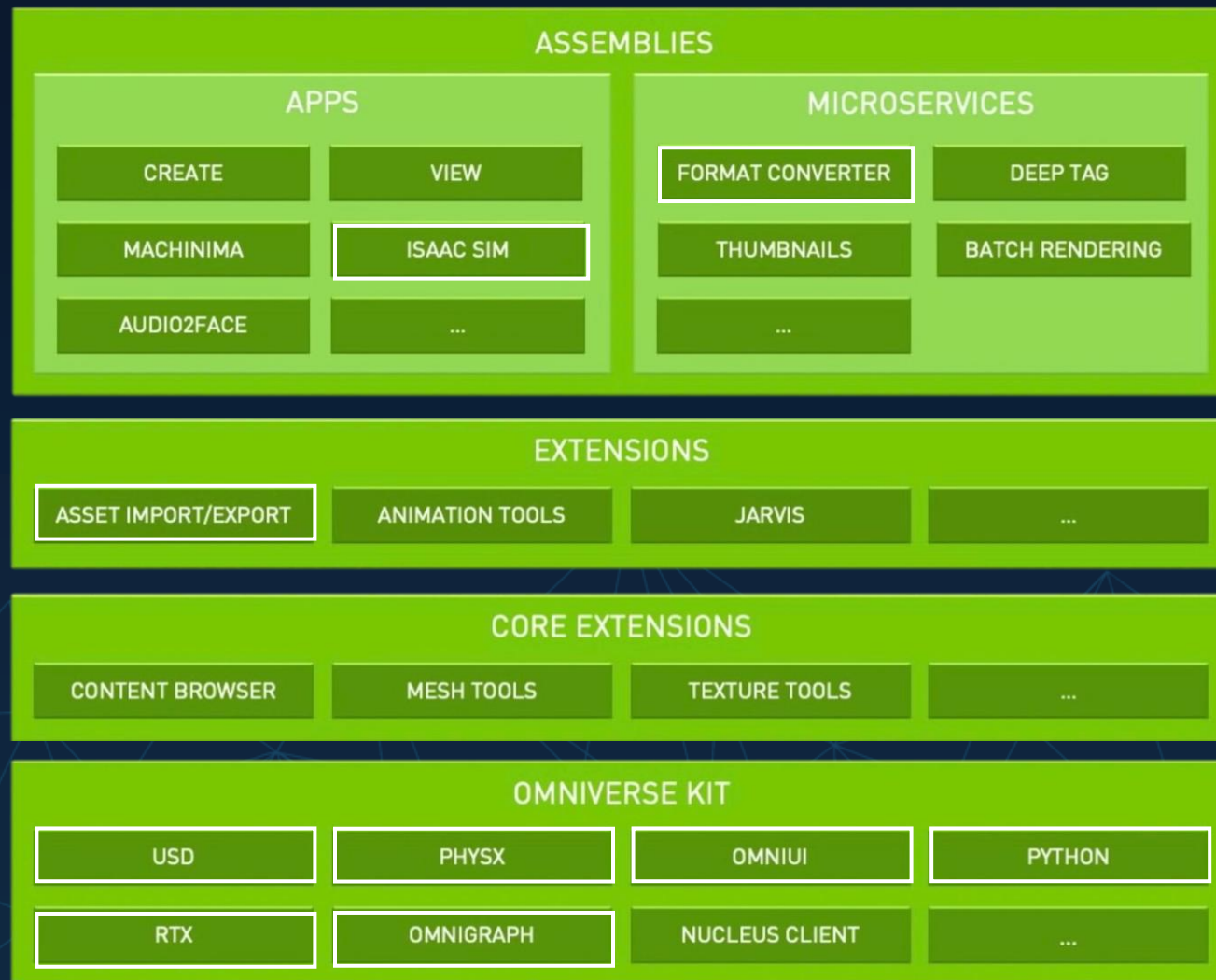
Introduction to Omniverse

Concept: Industrial Metaverse

- Virtual Environment
- Real-time Interaction and Collaboration
- Purpose can span the Design, Simulation, Observation or Replay of arbitrary Objects and Processes in any Industry
- The *NVIDIA Omniverse* platform is optimized to run 3D physics Simulations in real-time on *NVIDIA RTX* Hardware

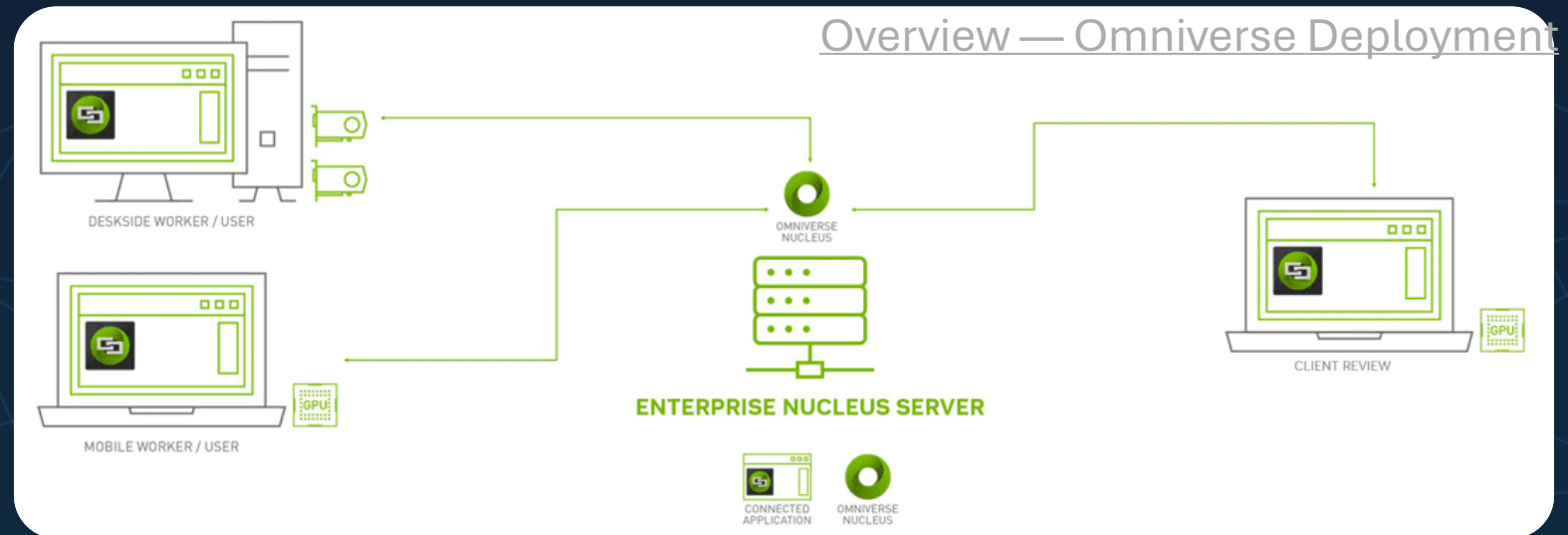


Omniverse Architecture



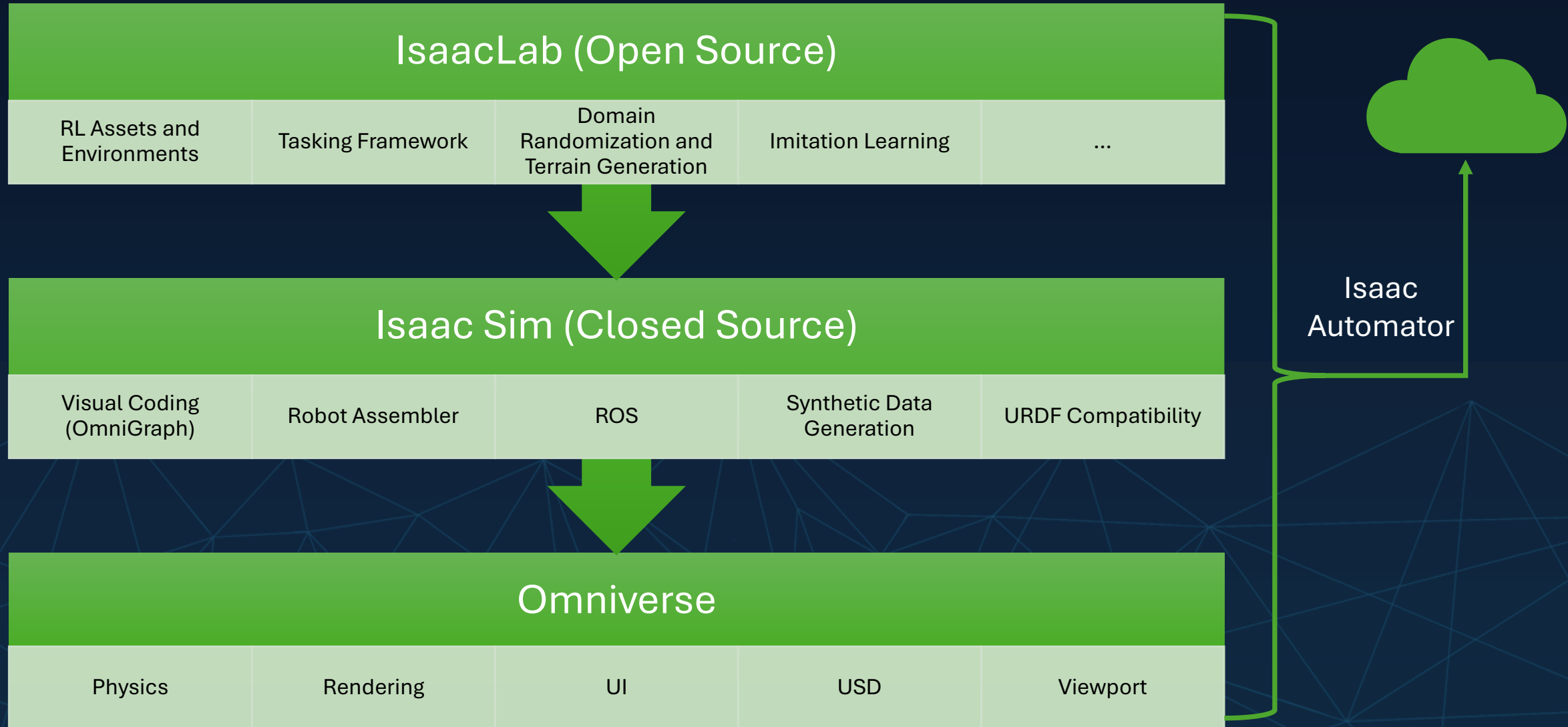
Nucleus – The Omniverse Database

- Enables Real-time Collaboration
- Comprehensive User/Permission Management
- Caching Mechanism for large Assets
- Wide API and Script Integration
- Enterprise Scalability





NVIDIA Isaac Ecosystem





Universal scene description



Interacting with USD (in Omniverse scene)

- UI Main Capabilities

- Viewport:
 - Scene
- Property Window:
 - Prim* Properties
- Omnigraph:
 - Process Logic
 - ROS Controller
- Various extensions

- API Main Capabilities

- Python interface
- Robot Controllers
- URDF import
- Create the scene for IsaacLab
- Configure the policy
- Start training

Robot Data formats

	URDF	SRDF	XRDF
Description Language	XML	XML	YAML
Purpose	Define Geometries and Kinematic Joints	Complements URDF Data with Semantic Grouping	Complements URDF Data with Kinematic Description
Collision Handling	Simple Shapes and Meshes	Collision Matrix (specifies handling for link pairs)	Collision Spheres along Robot Links
Joint Representation	Varying Types and Parameters	Defines Groups for Motion Planning	Advanced Configuration including Acceleration and Jerk Limits
Handling	Core Format in ROS and converted to USD upon import in Isaac Sim	ROS Moveit Assistant to generate Motion Plans	NVIDIA CuMotion and Lula to generate Motion Plans

Manual Modelling Phases Robot Simulation

1. Import URDF and convert to USD
2. Config Joint drives -> Test
3. Add Gripper/ Cameras
4. Add details to the stage of:
 1. URDF for ROS and RL
 2. SRDF for Moveit
 3. XRDF for Cumotion and LULA / RMPflow

NVIDIA PhysX

- The PhysX world comprises a collection of Scenes, each containing objects called Actors;
- Each Scene defines its own reference frame encompassing all of space and time;
- Actors in different Scenes do not interact with each other;
- Characters and vehicles are complex specialized objects made from Actors;
- Actors have a physical state: position and orientation, velocity or momentum, energy, etc;
- Actor physical state may evolve over time due to applied forces, constraints such as joints or contacts, and interactions between Actors.



Omniverse Stage



Until now:

- We reviewed necessary Omniverse Basics and Resources

5 min Break

To be continued:

- Practical session:
 - Extensions installation
 - Scene Manipulation
 - Robot import + tuning



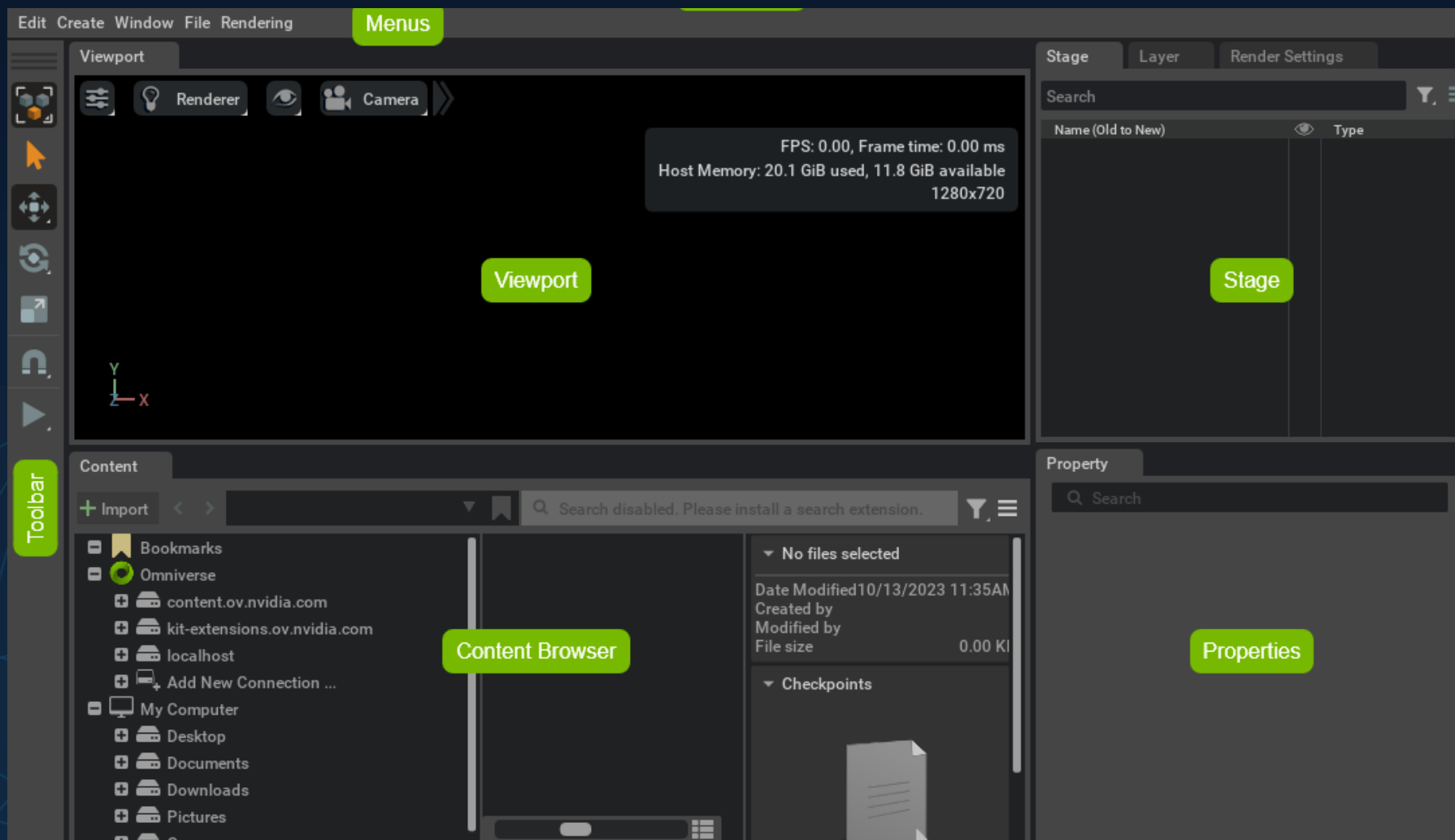
Hands on!

Getting started:

- Start Omniverse Isaac Sim on the Machine
 - Please give feedback if your application is loaded



Standard Layout

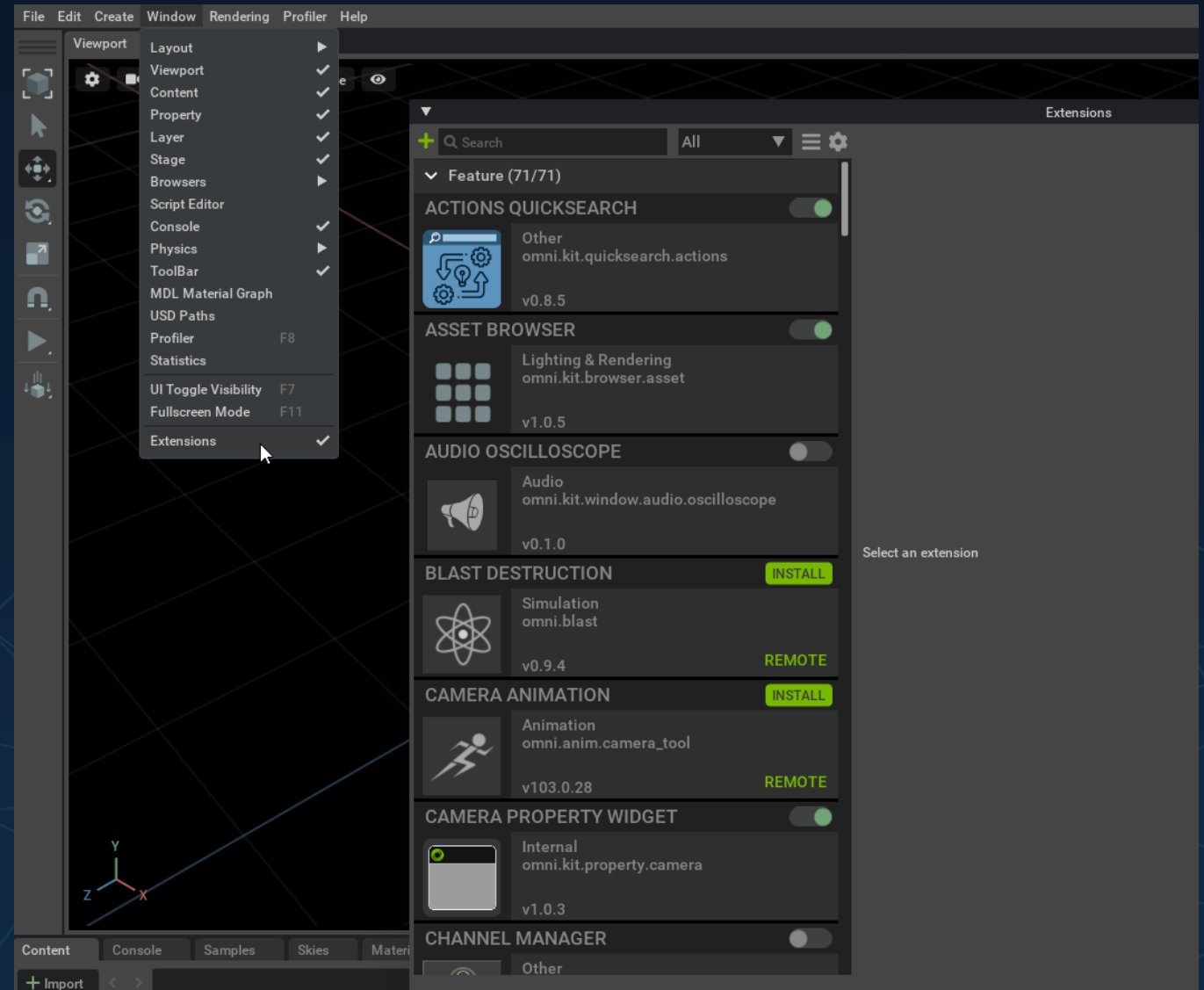


How-to: Viewport Navigation

- Orbit: Left mouse btn. + ALT
- Look: Right mouse btn. (RMB)
- Fly/ Walk: Hold RMB; *WASD* / Q up / E down
- Adjust Speed: Hold RMB; Scroll
- Jump to Selection: F

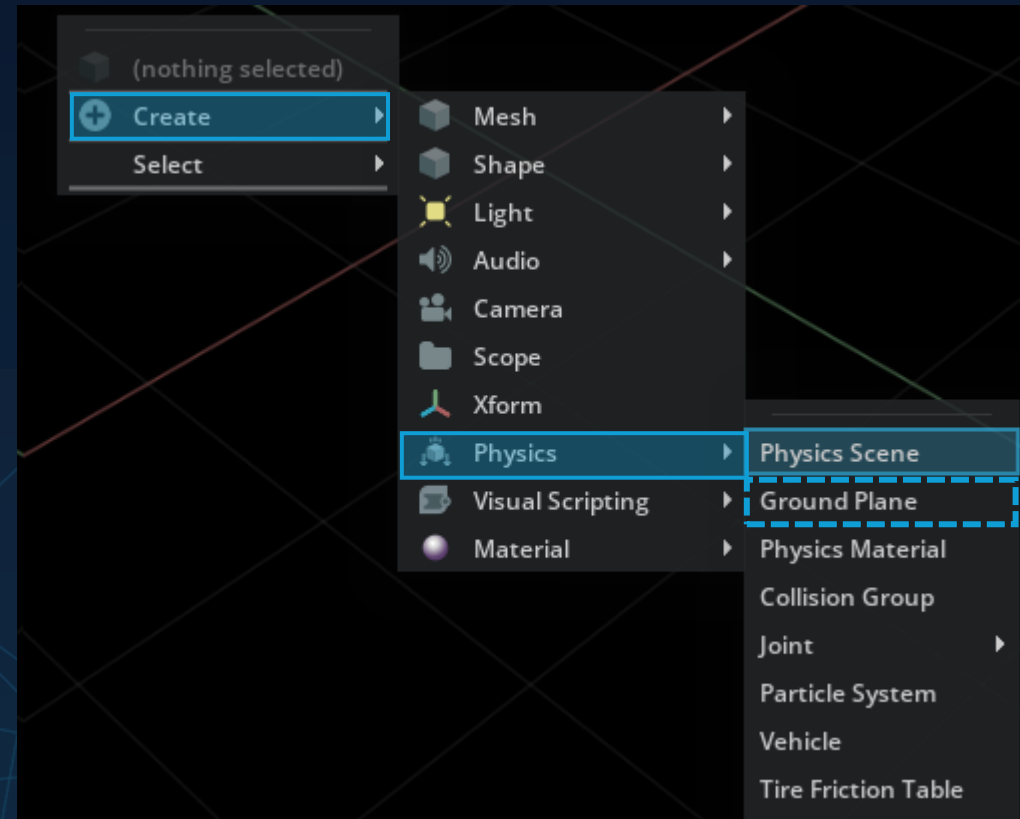
Scene Prep tools (extensions)

- Check for extensions:
 - Action Graph
 - URDF importer



How-to: Add Physics Scene

- Create a Physics Scene
- Create a ground plane

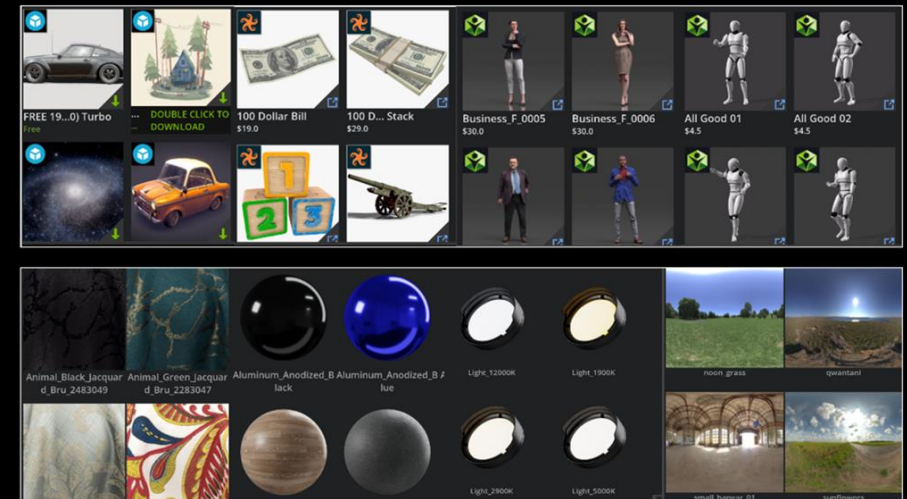


Asset Store

- Import a Box
 - Move above ground plane
 - Adjust the size
 - Add (Rigid Body and and Collider)
 - Press *Play* to start the Simulation

TYPES OF ASSETS

- 3D Models
- Motions/Animations
- Materials/Shaders
- Lights/Light Sets
- Environments/
Scenes/HDRIs
- Simulation-Ready
Assets
- Procedurally
Generated
Assets/Brushes
- Volumetric Effects e.g.
Smoke, fog



How-to: Use Asset Store



1. Search
2. Drag & Drop
(to Stage or Viewport)

a

b

Isaac Sim Asset Store

Box

1

2

003_cracker_box 003_cracker_box 004_sugar_box 004_sugar_box 008_pudding_box

009_gelatin_box sm_white_box sm_black_box

File name: 003_cracker_box.usd
File size: 460 KB
File Path: Props/VCB/Axis_Aligned/003_cracker_box.usd

Stage

Layer

Render Settings

Search

Name (Old to New)	Type
World (defaultPrim)	Xform
PhysicsScene	PhysicsScene
GroundPlane	Xform
Environment	Xform

Property

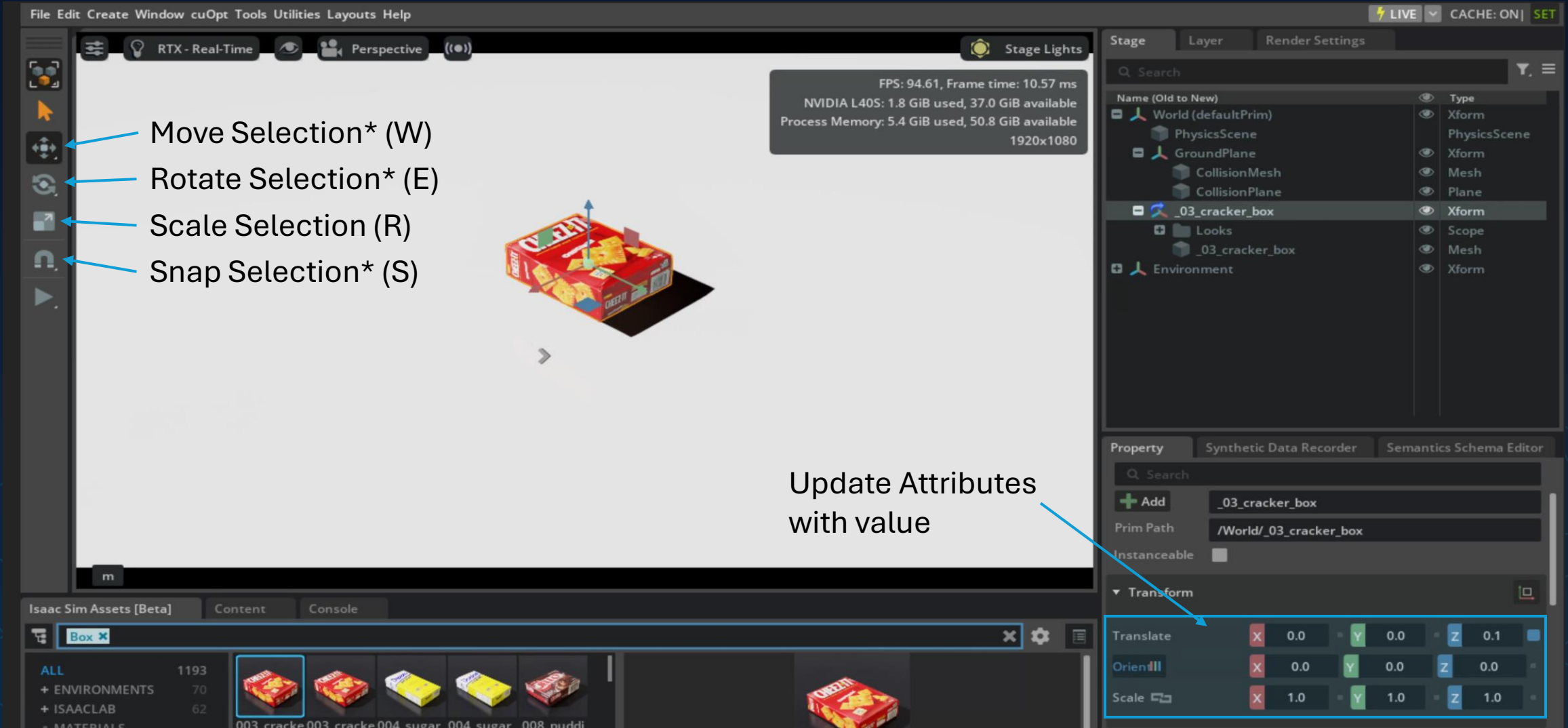
Synthetic Data Recorder

Semantics Schema Editor

Search

FPS: 118.36, Frame time: 8.45 ms
NVIDIA L40S: 1.6 GiB used, 37.1 GiB available
Process Memory: 5.4 GiB used, 50.7 GiB available
1920x1080

How-to: Manipulation in Viewport

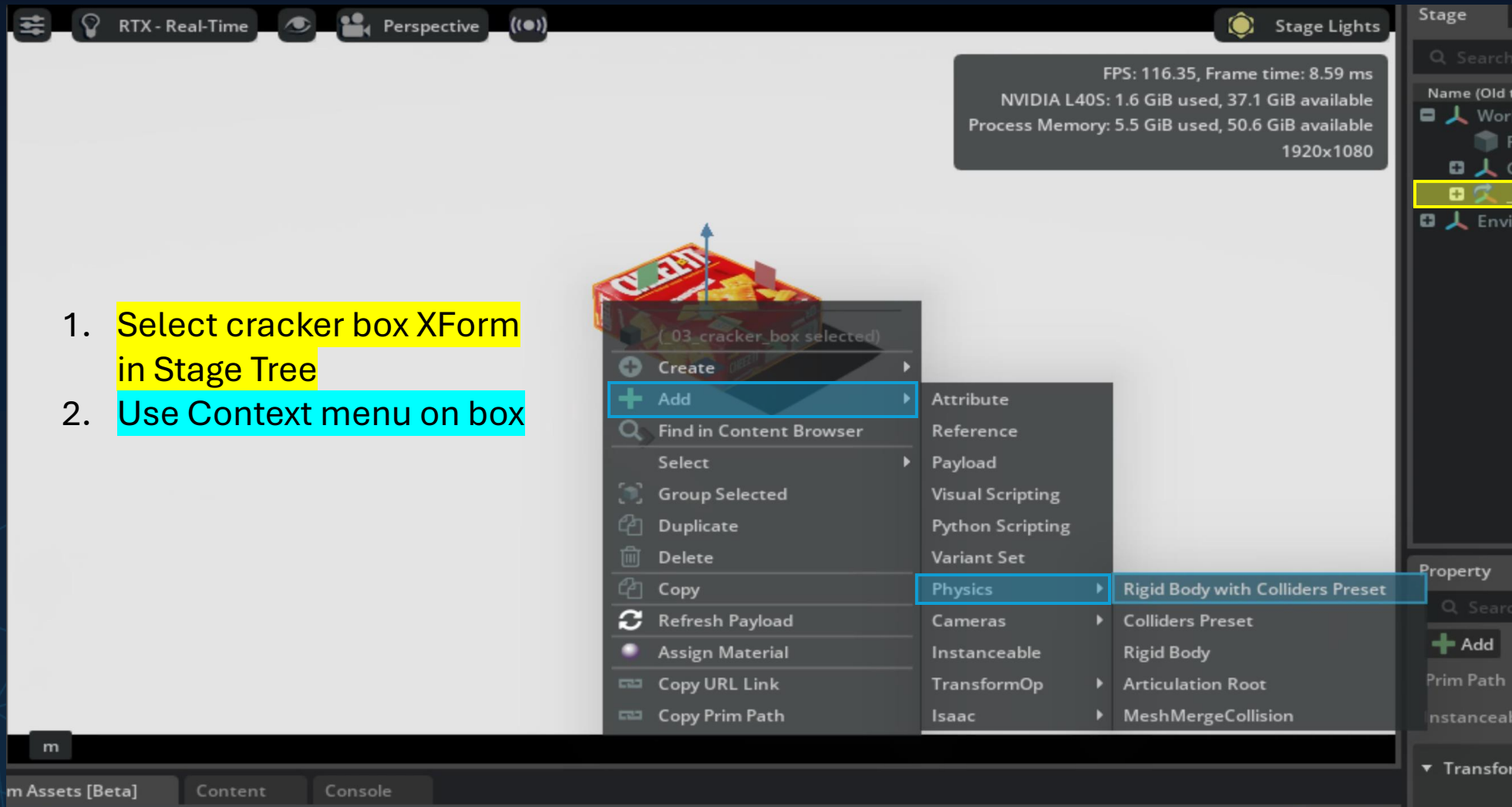


The screenshot displays the Isaac Sim interface with the following components:

- Top Bar:** File Edit Create Window cuOpt Tools Utilities Layouts Help. Modes: RTX - Real-Time, Perspective, Stage Lights.
- Left Panel:** Manipulation tools: Move Selection* (W), Rotate Selection* (E), Scale Selection (R), Snap Selection* (S).
- Viewport:** A 3D scene showing a red cracker box. A status box displays: FPS: 94.61, Frame time: 10.57 ms, NVIDIA L40S: 1.8 GiB used, 37.0 GiB available, Process Memory: 5.4 GiB used, 50.8 GiB available, 1920x1080.
- Right Panel:** Stage, Layer, Render Settings tabs. A search bar and a list of objects: World (defaultPrim), PhysicsScene, GroundPlane, CollisionMesh, CollisionPlane, _03_cracker_box, Looks, _03_cracker_box, Environment.
- Bottom Panel:** Isaac Sim Assets [Beta], Content, Console. A list of assets: Box, 003 cracke, 003 cracke, 004 sugar, 004 sugar, 008 puddi.
- Property Panel:** Synthetic Data Recorder, Semantics Schema Editor. A search bar and a list of properties: Add, _03_cracker_box, Prim Path, /World/_03_cracker_box, Instanceable, Transform.
- Transform Panel:** Translate, Orient, Scale. Values: X: 0.0, Y: 0.0, Z: 0.1; X: 0.0, Y: 0.0, Z: 0.0; X: 1.0, Y: 1.0, Z: 1.0.

Update Attributes with value

How-to: Add Rigid-body & Collider



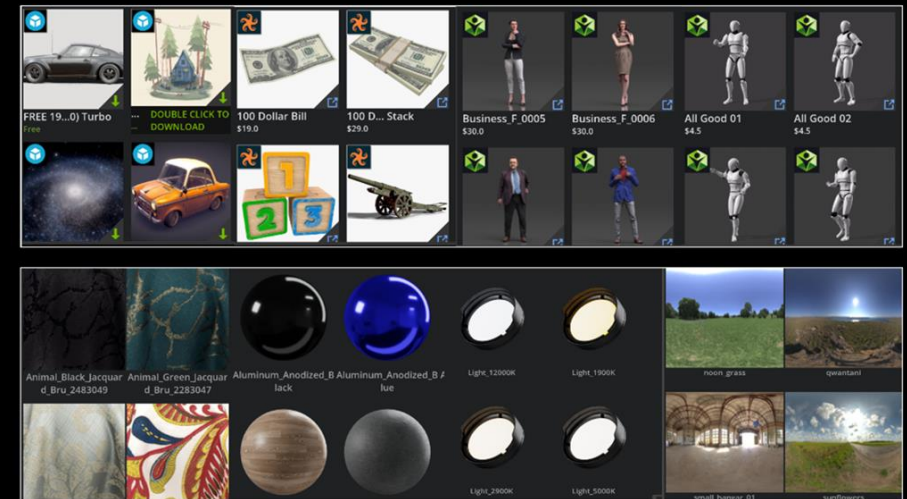
Asset Store



- Import a Box
 - Move above ground plane
 - Adjust the size
 - Add (Rigid Body and and Collider)
 - Press *Play* to start the Simulation

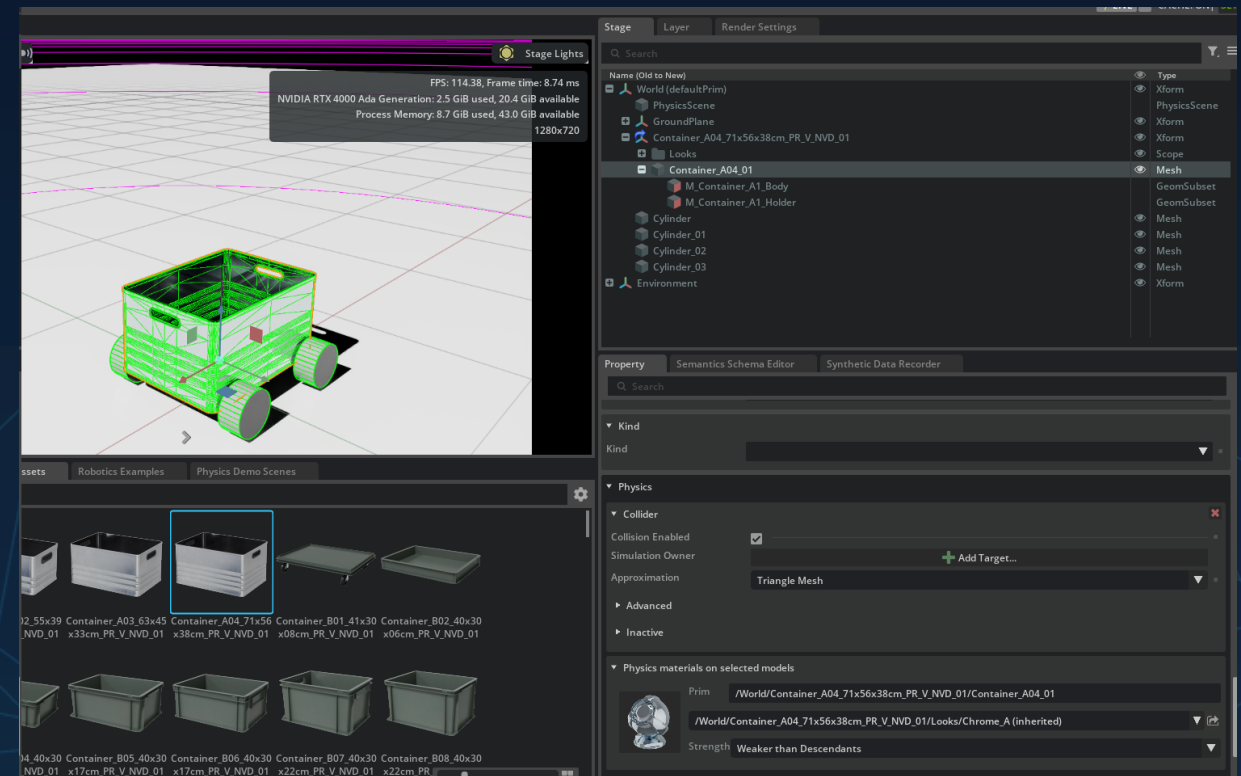
TYPES OF ASSETS

- 3D Models
- Motions/Animations
- Materials/Shaders
- Lights/Light Sets
- Environments/
Scenes/HDRIs
- Simulation-Ready
Assets
- Procedurally
Generated
Assets/Brushes
- Volumetric Effects e.g.
Smoke, fog

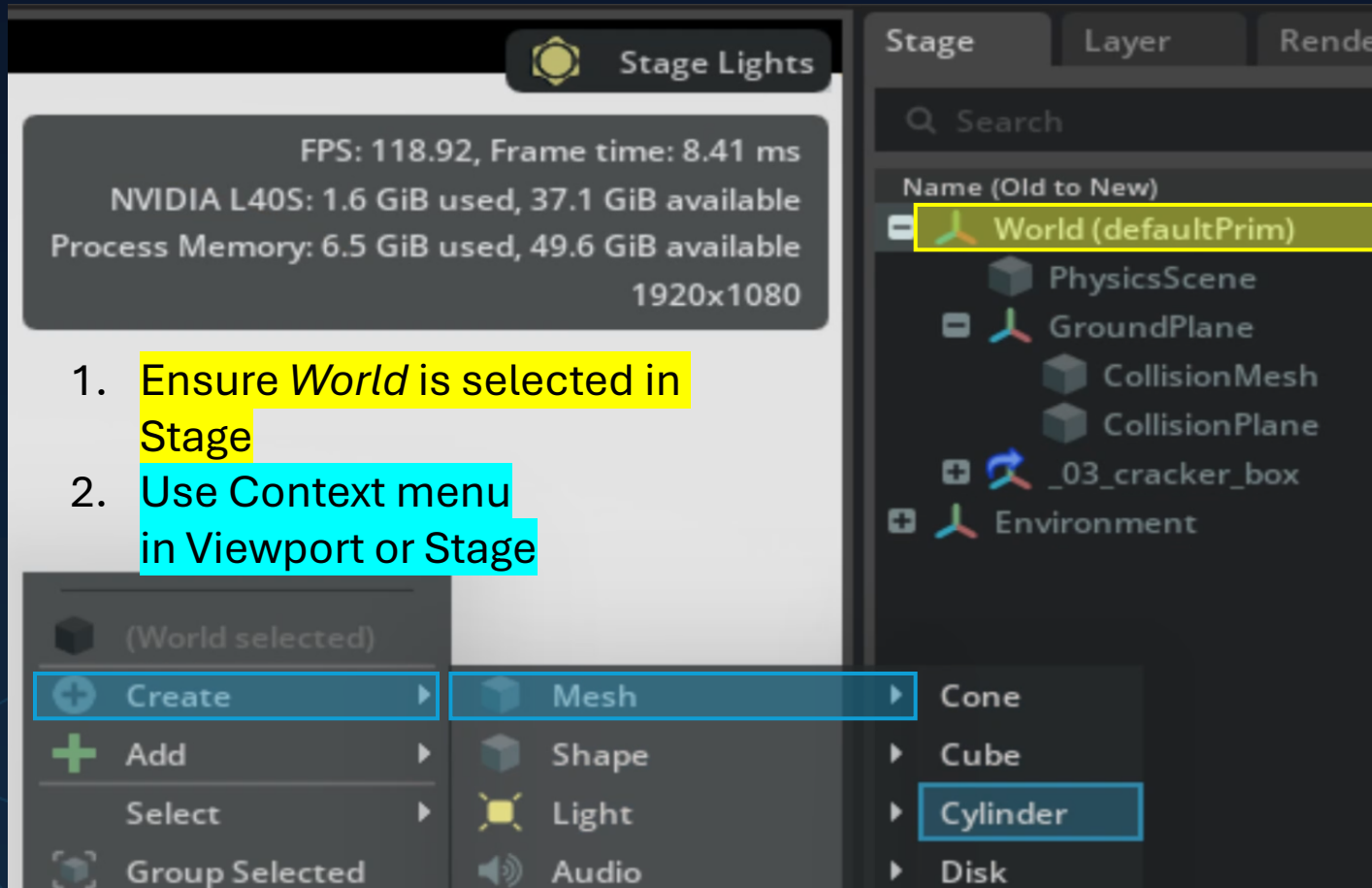


Assemble the first self-built Robot

- Stop the Simulation
- Create Cylinders
 - Move/ orient to the side
 - Create a revolute joint
 - Add Angular drive
 - Modify target velocity + stiffness
- Start the Simulation



How-to: Add Cylinder



1. Ensure *World* is selected in Stage
2. Use Context menu in Viewport or Stage

3. Adjust scaling and orientation to represent a wheel
4. Add *Rigid Body & Collider*
5. Use *Ctrl+C* and *Ctrl+V* to replicate the wheel
6. Move the wheels to reasonable positions around the box

How-to: Add Joint



NVIDIA L40S: 1.6 GiB used, 37.1 GiB available
Process Memory: 6.7 GiB used, 49.4 GiB available
1920x1080

1. Holding *Ctrl*, select the box then the cylinder
2. Create joint using the context menu
3. Repeat for all *wheels*

World (defaultPrim)
PhysicsScene
GroundPlane
CollisionMesh
CollisionPlane
_03_cracker_box
Looks
_03_cracker_box
Cylinder
Cylinder (2 models selected)
Cylinder (Cylinder hovered)

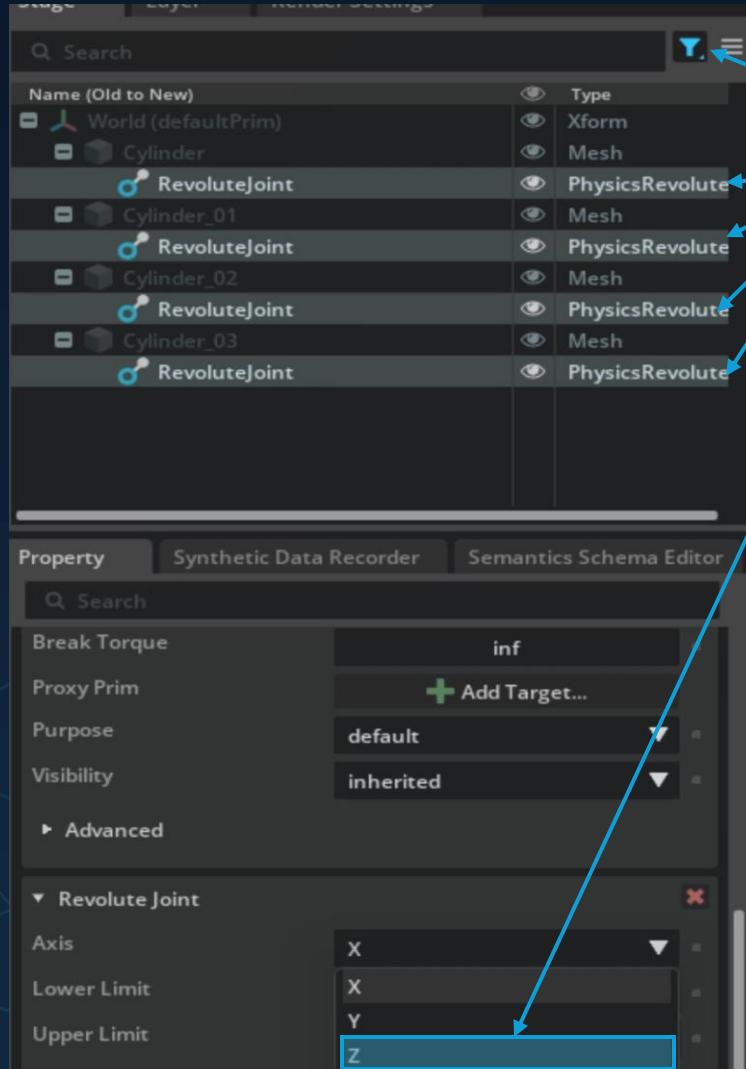
Mesh
Shape
Light
Audio
Camera
Scope
Xform
Visual Scripting
Physics
Material

Create
Add
Group Selected
Duplicate
Delete
Save Selected
Deactivate

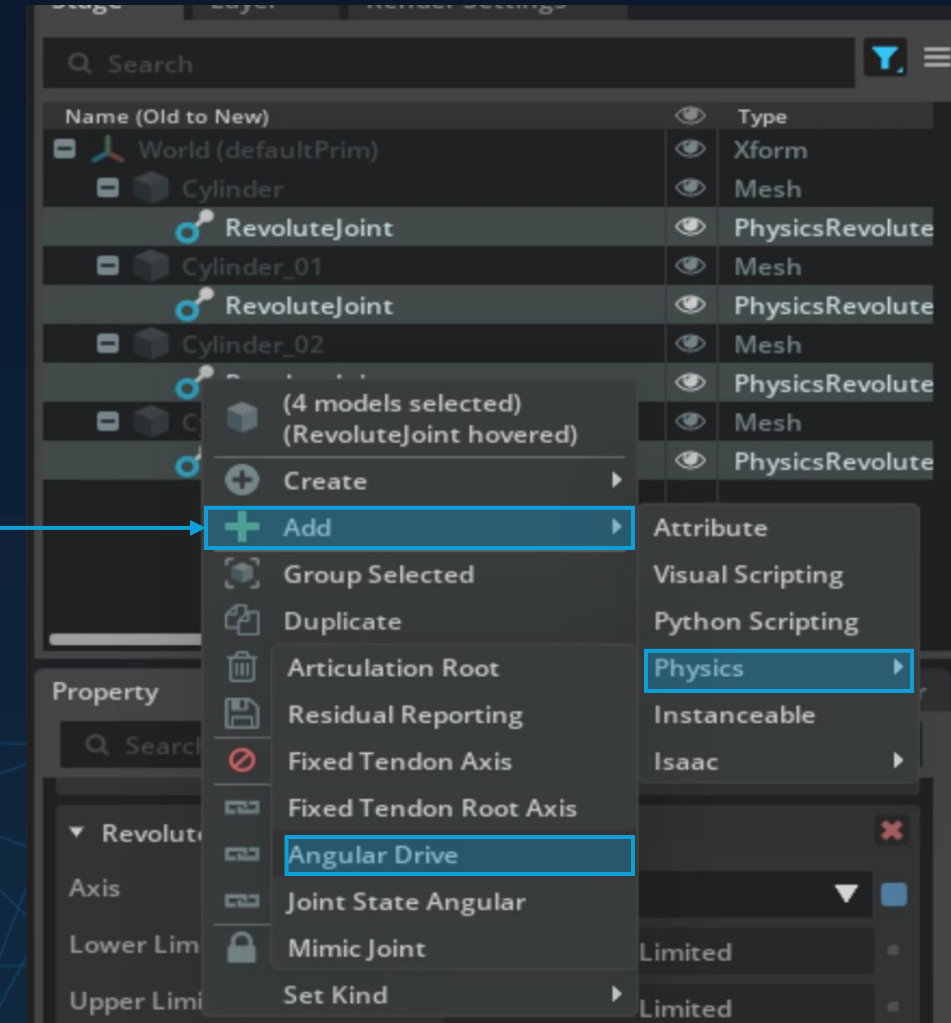
Physics Scene
Ground Plane
Physics Material
Collision Group
Joint
Attachment
Particle System
Vehicle

D6 Joint
Fixed Joint
Revolute Joint
Prismatic Joint

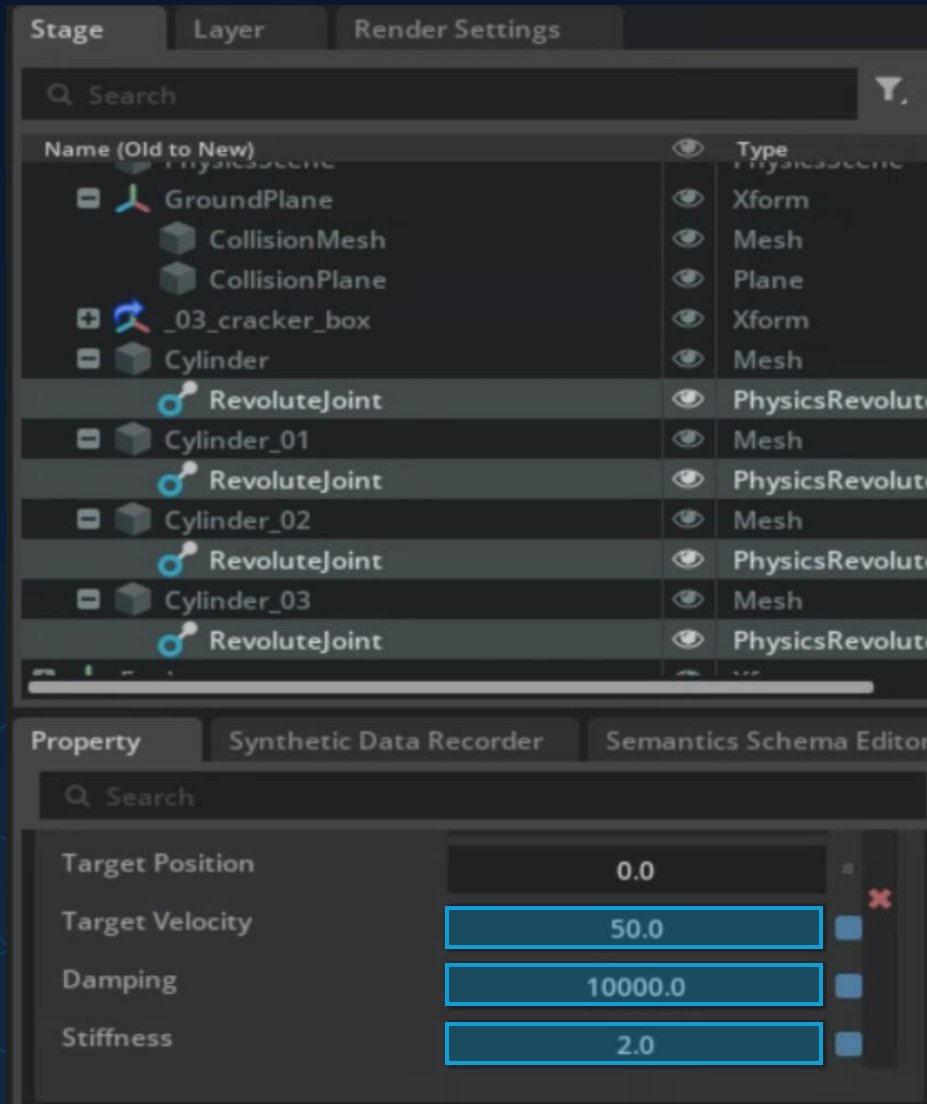
How-to: Adjust Joint and Add Drive



1. Filter for Joints
2. Select all joints
(via *Shift* or *Ctrl*)
3. Adjust joint rotation
4. Add Drives to joints
(via Context menu)



How-to: Adjust Drive

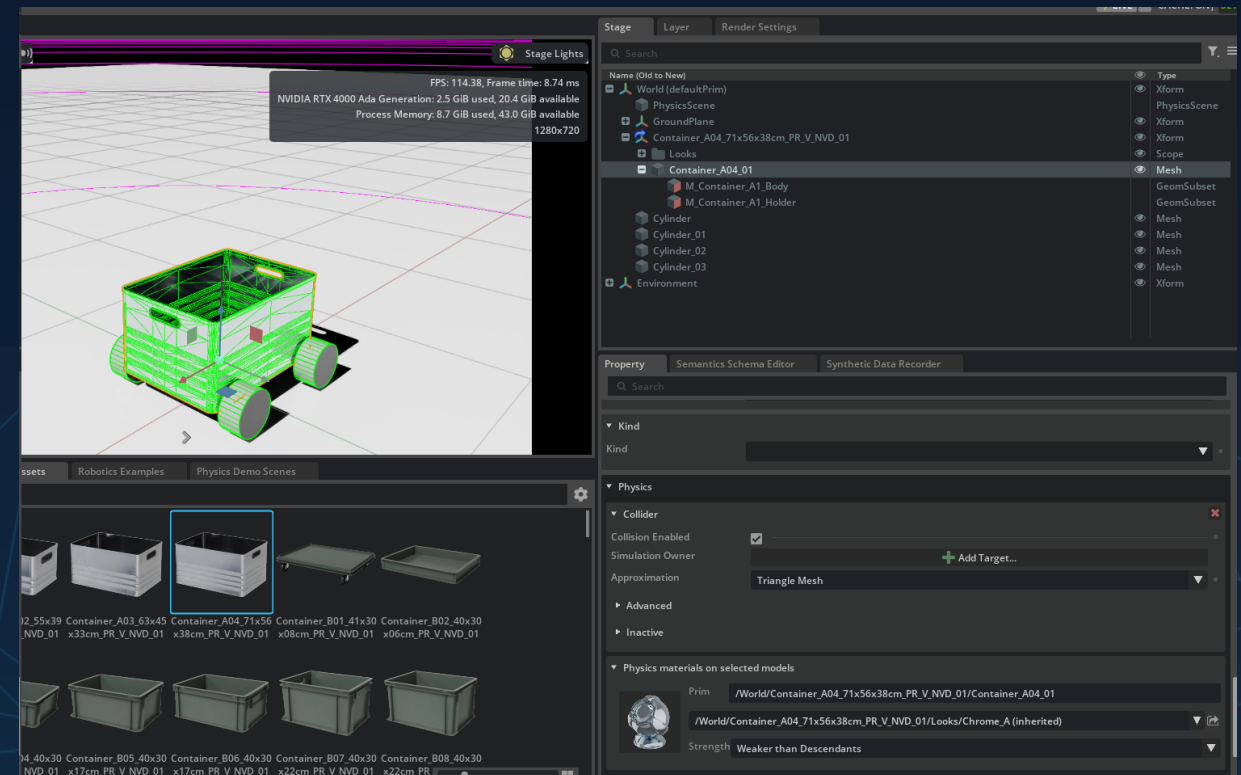


1. Set a Velocity(~20-100)
2. Iterate:
 1. Set Damping & Stiffness Values
 2. Restart Simulation for Testing*

*Not restarting the simulation leads to misleading results due to the forces that already applied before the update

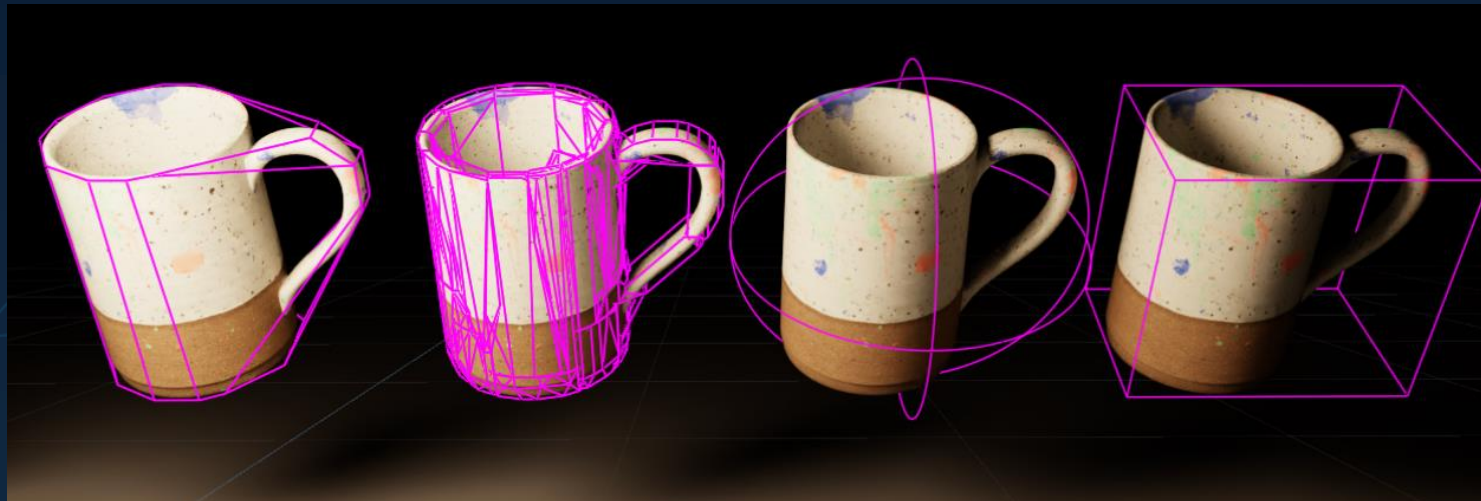
Assemble the first self-built Robot

- Stop the Simulation
- Create Cylinders
 - Move/ orient to the side
 - Create a revolute joint
 - Add Angular drive
 - Modify target velocity + stiffness
- Start the Simulation

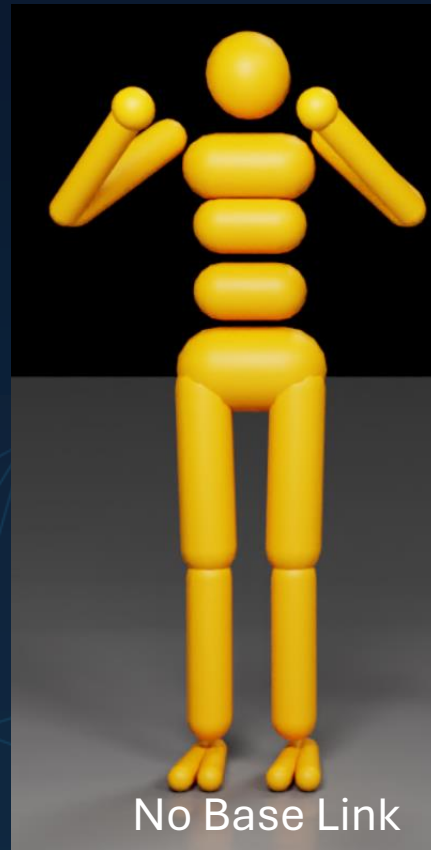


Rigid Bodies and Colliders

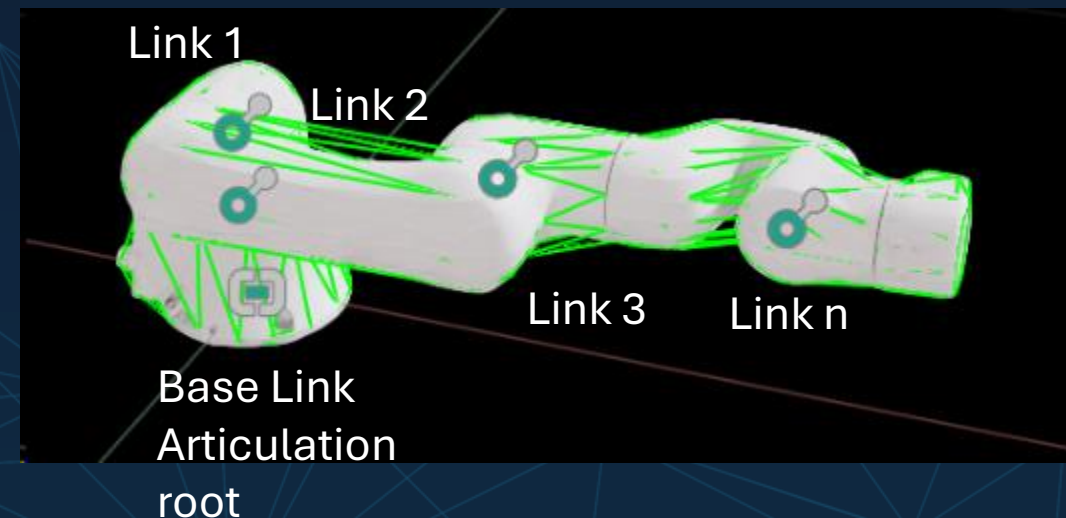
- Simpler approximation = faster Simulation



Robot assembly

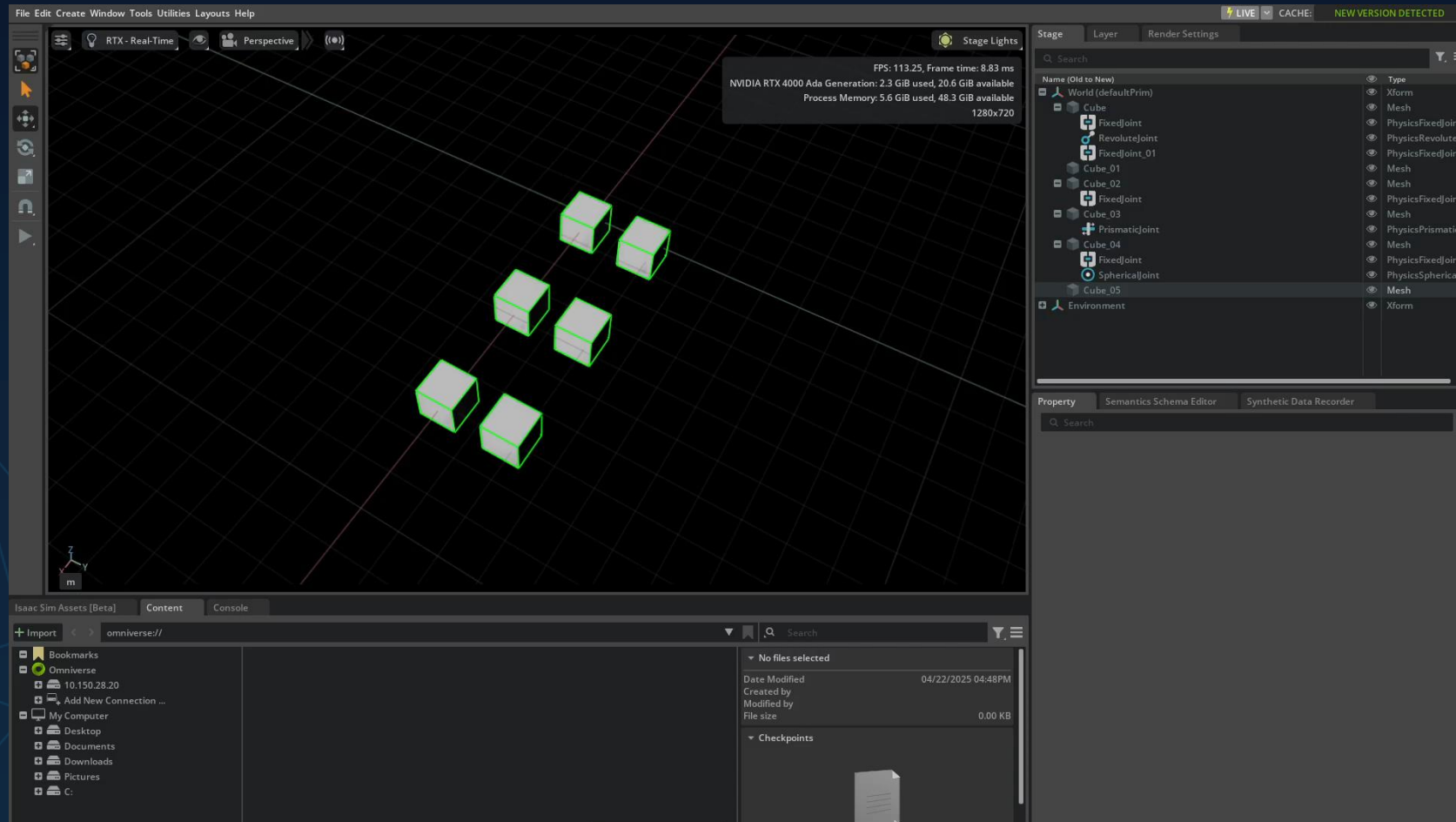


- Structure of mobile / static robot
 - Articulation chain
 - root link free / fixed



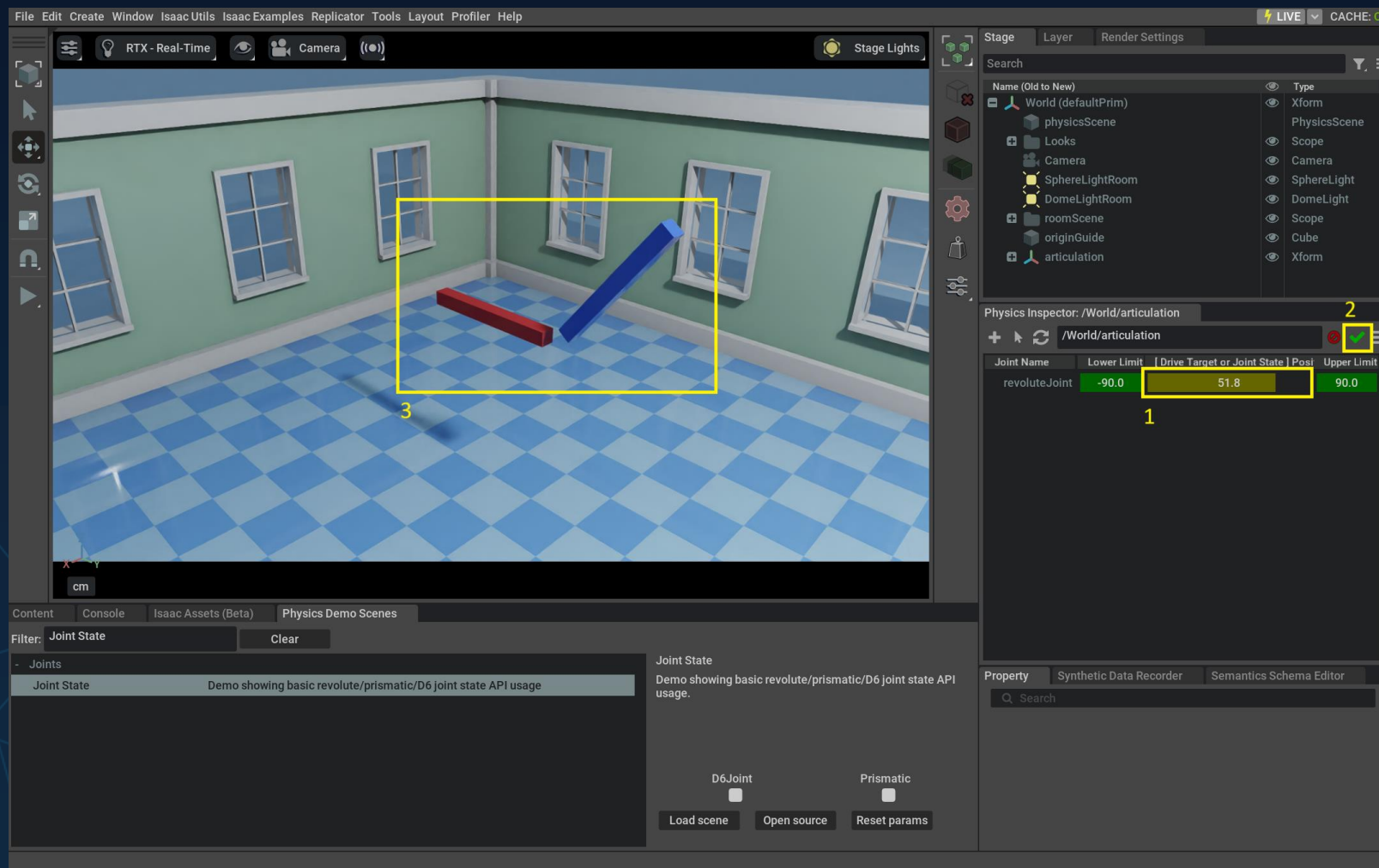


Joint types and properties

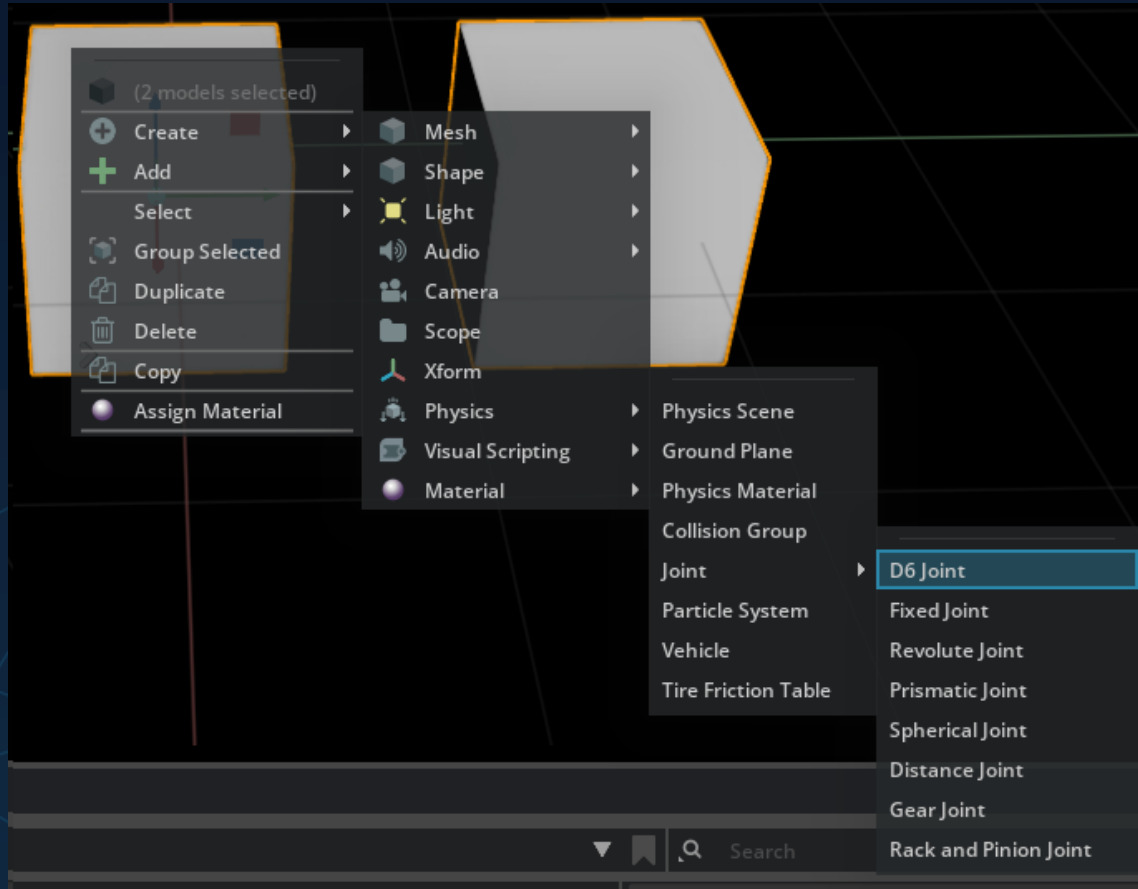




Joints



Joint types and properties



Joint drives are internally simulated with PD-Controllers

- *Stiffness* property \leftrightarrow Proportional coefficient
- *Damping* property \leftrightarrow Derivative coefficient

⚠ This can lead to unexpected behavior at low simulation frequencies

Until now:

- We reviewed necessary Omniverse Basics and Resources
- Practical session:
 - Extensions installation
 - Scene Manipulation

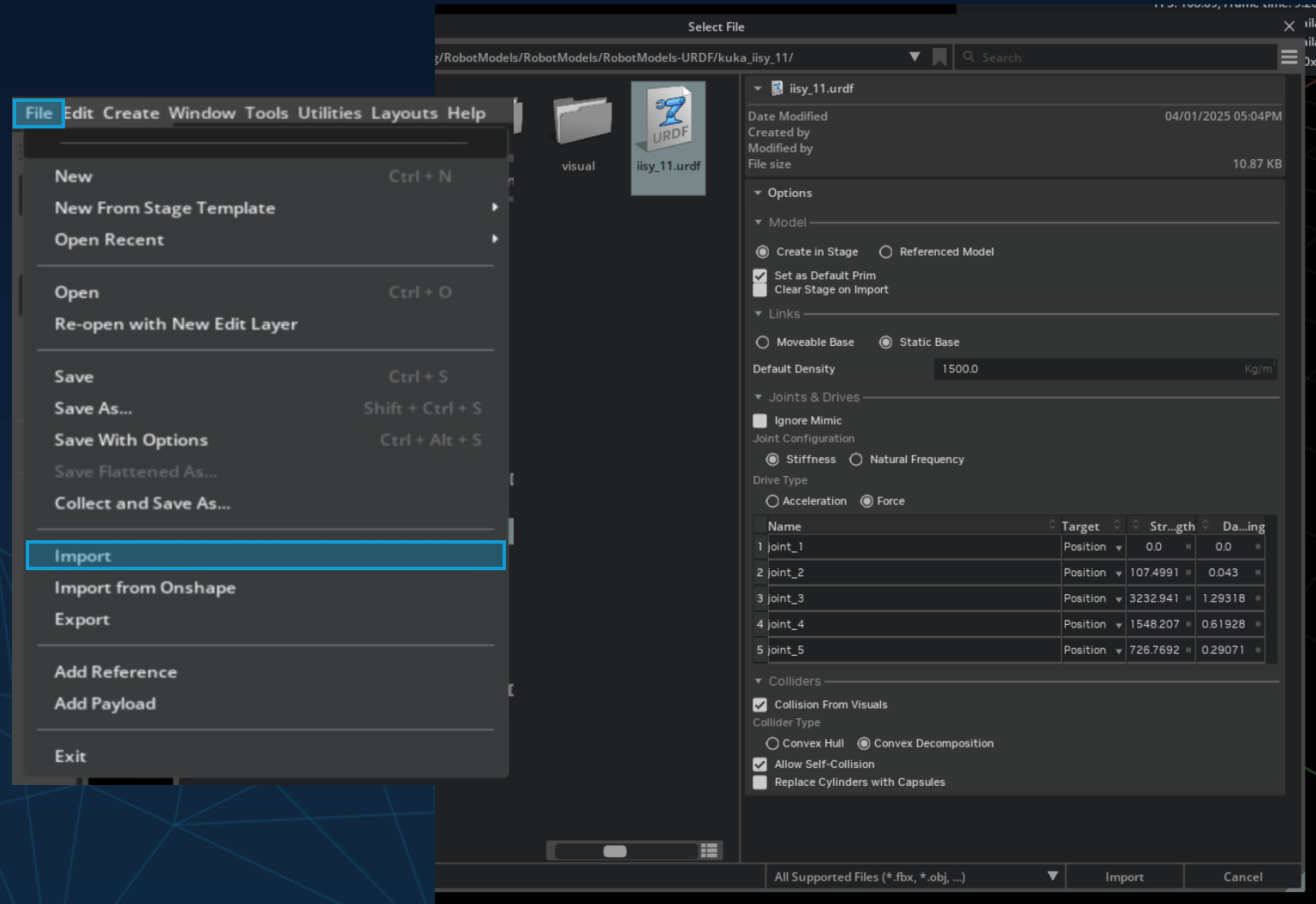
10 min Break

To be continued:

- Practical session:
 - Robot import + tuning

How-to: Starting with a real Robot

- Open *File*->*import*
- Navigate to a URDF File
- Modify the Options
 - Verify *Model* and *Links*
 - Adjust *Joints & Drives*
 - Verify *Colliders*
- Click *Import*





Robot import Tools

- LULA URDF Import Tool:
 - Import Robot URDF
 - Start scene -> look at joint drives
- Physics Inspector Tool:
 - Activate scene
 - Choose IISY to inspect-> control joint limits/ direction
- Gain tuner:
 - Adjust Stiffness & Damping and test the parameters
- Preparation of self and external collision shape:
 - Sphere approximation Collider tool for RMP

Physics Inspector: /lbr_iisy3_r760/Robotiq_2F_140_p...

+ ↩ ↺ /lbr_iisy3_r760/Robotiq_2F_140_physics_edit

Joint Name	Lower Limit	[Drive Target or Joir	Upper Limit
left_inner_knuckle_joint	Set Limit	0.0	Set Limit
right_inner_knuckle_joint	Set Limit	0.0	Set Limit

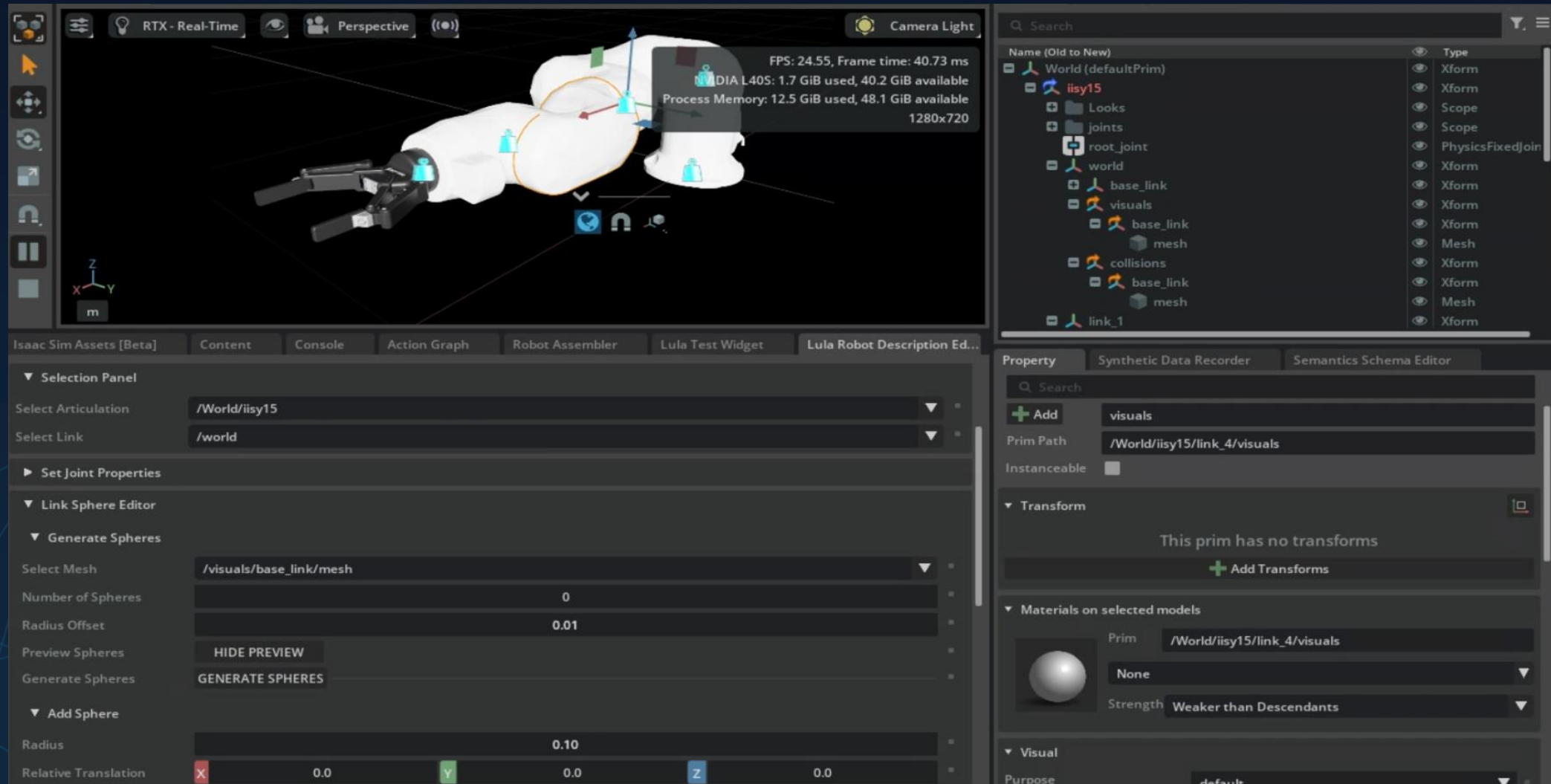
▼ Robotiq_2F_140_physics_edit

Joint Name	Lower Limit	[Drive Target or Joir	Upper Limit
joint_1	-185.0	-93.5	185.0
joint_2	-230.0	-77.0	50.0
joint_3	-150.0	69.6	150.0
joint_4	-175.0	16.2	175.0
joint_5	-110.0	64.6	110.0
joint_6	-220.0	-220.0	220.0
FixedJoint		Fixed Joint	
finger_joint	0.0	0.0	45.0
left_outer_finger_joint	0.0	0.0	180.0
left_inner_finger_joint	Set Limit	-42.7	Set Limit
left_inner_finger_pad_joint	-45.0	-45.0	45.0
right_outer_knuckle_joint		Mimic Joint	
right_outer_finger_joint	0.0	0.0	180.0
right_inner_finger_joint	Set Limit	-45.0	Set Limit
right_inner_finger_pad_joint	-45.0	0.0	45.0

Robot Description Editor

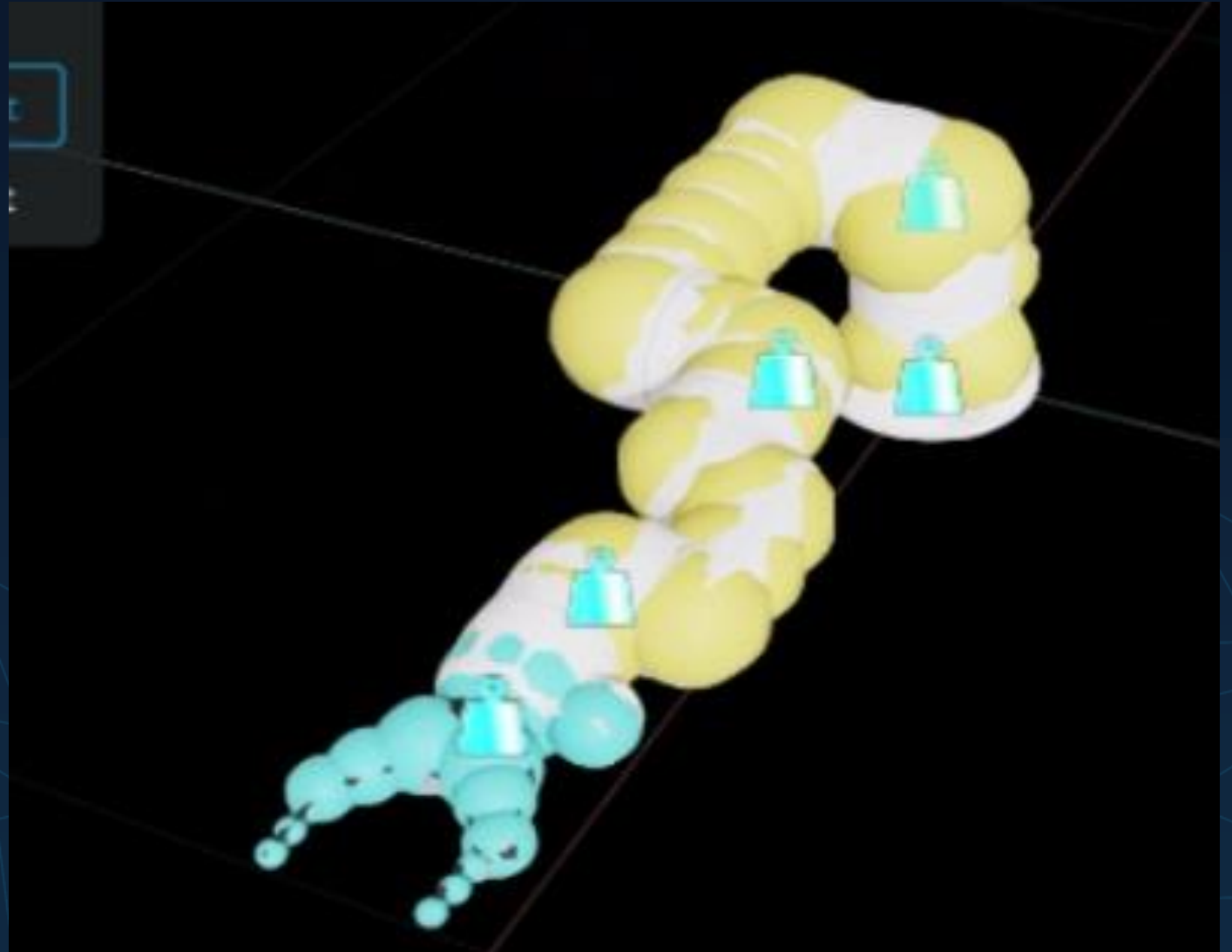


IPI
Institut für Produktion
und Informatik





Sphere Approximation Result



Lula Test Widget



The screenshot displays the Lula Test Widget interface within a 3D simulation environment. The main window shows a 3D view of a robot model (Lula) with various colored spheres representing collision points. The interface includes several panels and tabs:

- Top Bar:** RTX - Real-Time, Perspective, Default.
- Top Right Panel:** Search bar, Name (Old to New), Type, and a list of objects: World (defaultPrim), LulaRobotDescriptionEditor, iisy15, Looks, Lula, Trajectory, Target_1, Target_2, Target_3, Target_4.
- Bottom Left Panel:** Overview, Selection Panel, Lula Kinematics Solver, Lula Trajectory Generator, Custom Trajectory Panel.
- Bottom Right Panel:** Property, Synthetic Data Recorder, Semantics Schema Editor.

Selection Panel Details:

- Select Articulation: /World/iisy15/root_joint
- Robot Description YAML: /home/omniverse-1/Documents/Workshop/iisy15.yaml
- Robot URDF: /home/omniverse-1/Documents/RobotModels/RobotModels-URDF/kuka_iisy_15/iisy_15.urdf
- Load Selected Config: LOAD
- Select End Effector Frame: tool0
- Use Orientation Targets: ☒
- Visualize End Effector Pose: ☒

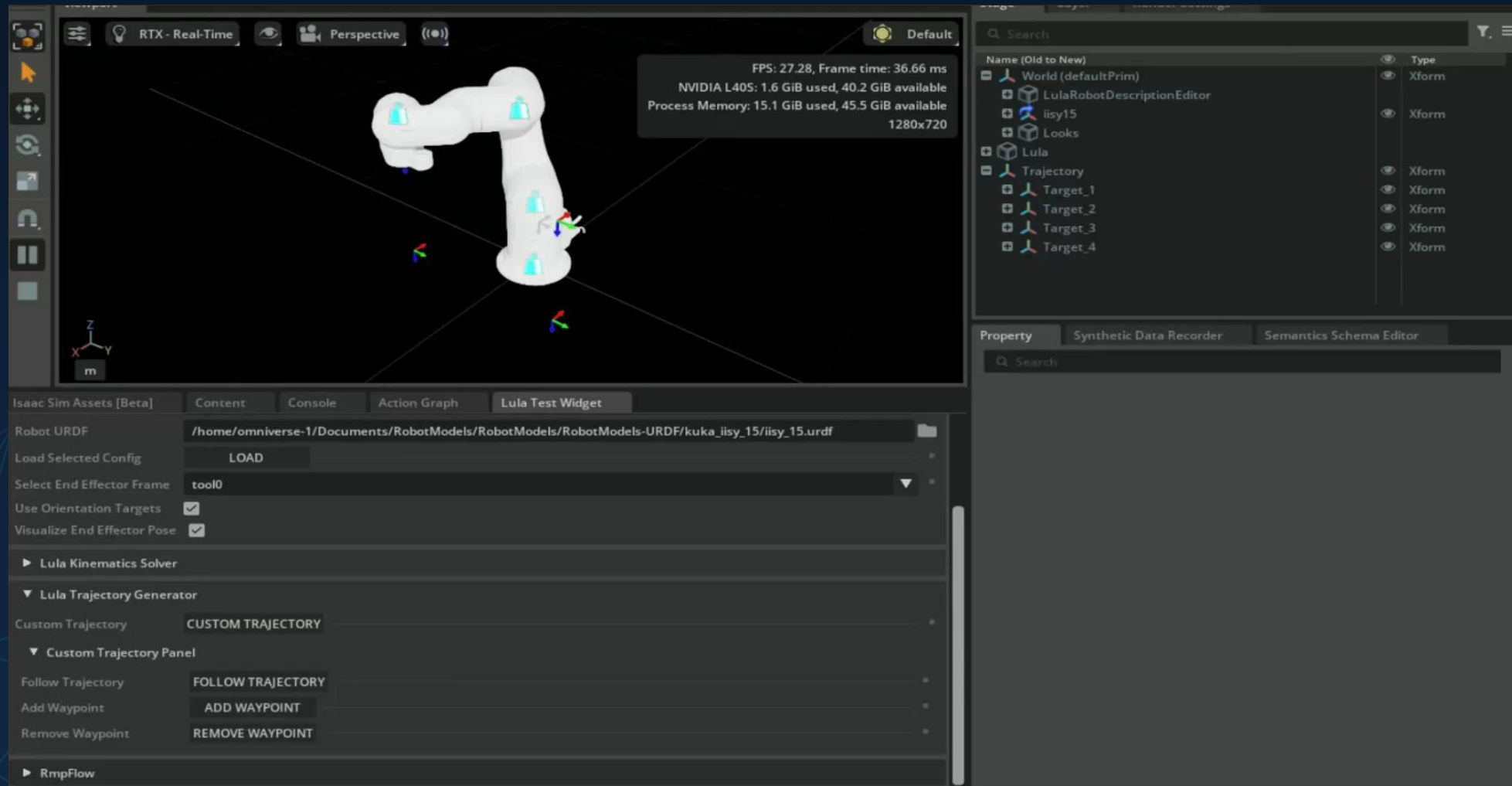
Lula Trajectory Generator Details:

- Custom Trajectory: CUSTOM TRAJECTORY
- Follow Trajectory: FOLLOW TRAJECTORY

Property Panel Details:

- Search: (4 models selected) common attributes shown
- Prim Path: Mixed
- Instanceable: ☐
- Transform: Translate (X: 0.3, Y: Mixed, Z: Mixed), Orientation (X: -180.0, Y: 0.0, Z: -180.0), Scale (X: 1.0, Y: 1.0, Z: 1.0)
- Materials on selected models: Prim (Mixed), None, Strength (Weaker than Descendants)
- Visual: Purpose (default)

Lula Test Widget



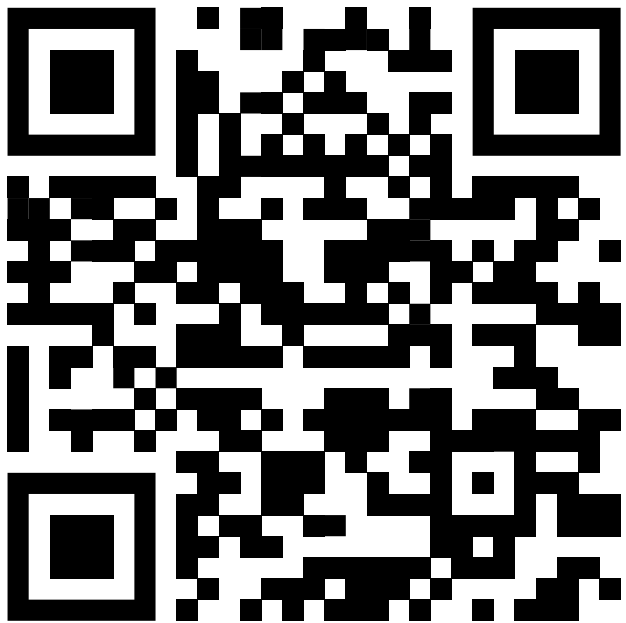


Sensors

- Adding a Wrist Camera to the Robot
 - Add the Camera under the last Robot arm link rename it to wrist camera
 - Open a second viewport and select the Wrist camera

End of Day 1

Please provide some
Feedback
Verbal or via:



For further questions mail us:
fabian.fichtl@hs-kempten.de
julian.zuern@hs-kempten.de