IPI
Institut für Produktion und Informatik

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

IPI
Institut für Produktion
und Informatik

Trainer Introduction

Fabian Fichtl

Julian Zürn

IPI
Institut für Produktion und Informatik

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

HIGHTECH Agenda Bayern

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:

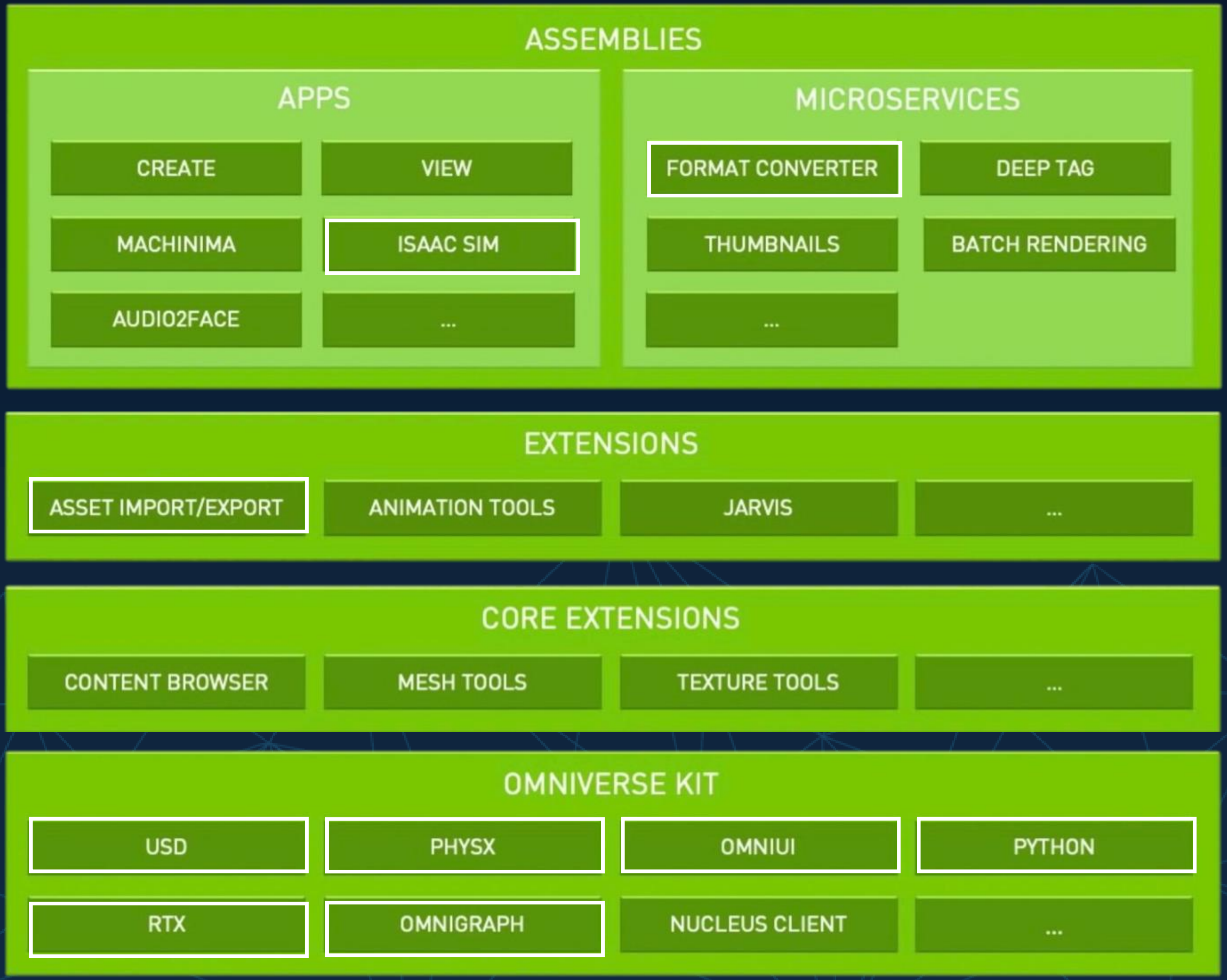
- Debugging Tools:
  - Console, VS Code, UI
- Scene Architecture
  - Build your Scenario: Robot, Gripper, Controller, Environment, Handle Object
- Pick & Place
- Information Channels | Further Courses

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

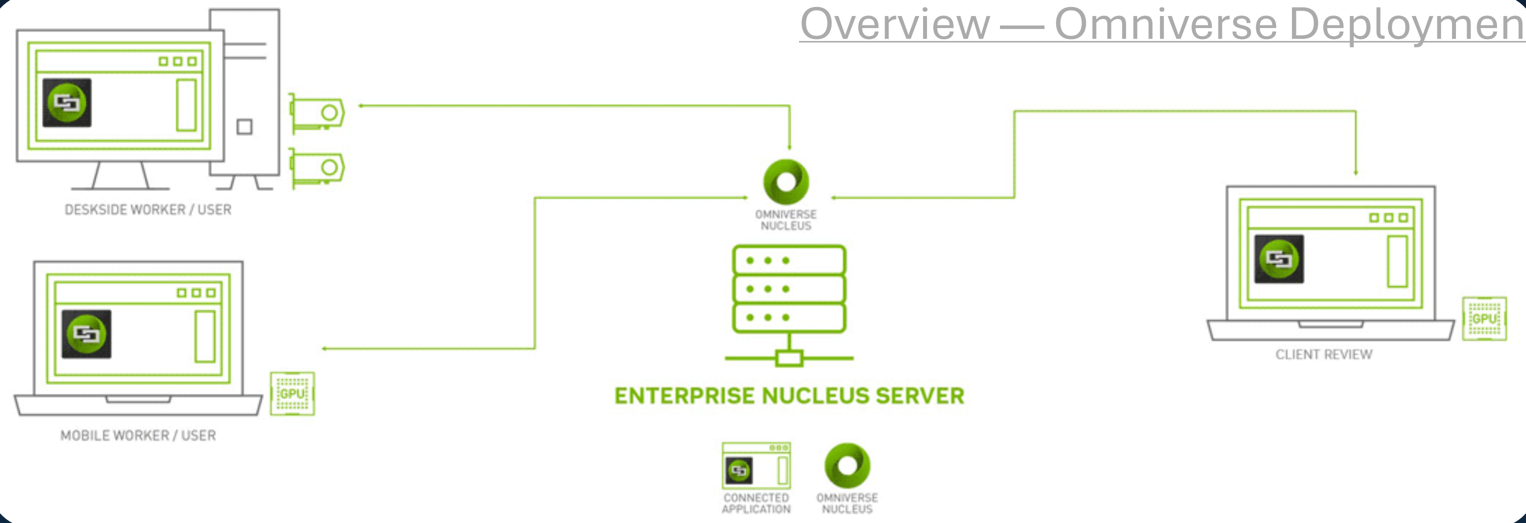
Omniverse Architecture



Nucleus – The Omniverse Database

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

Overview — Omniverse Deployment



NVIDIA Isaac Ecosystem

IPI Institut für Produktion und Informatik

IsaacLab (Open Source)

RL Assets and Environments	Tasking Framework	Domain Randomization and Terrain Generation	Imitation Learning	...

Isaac Sim (Closed Source)

Visual Coding (OmniGraph)	Robot Assembler	ROS	Synthetic Data Generation	URDF Compatibility

Omniverse

Physics	Rendering	UI	USD	Viewport

Isaac Automator

Universal scene description

Interacting with USD (in Omniverse scene)

IPI
Institut für Produktion und Informatik

- 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

*Nodes in the USD-scene tree are called Primitives or Prims

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	NVIDA CuMotion and Lula to generate Motion Plans

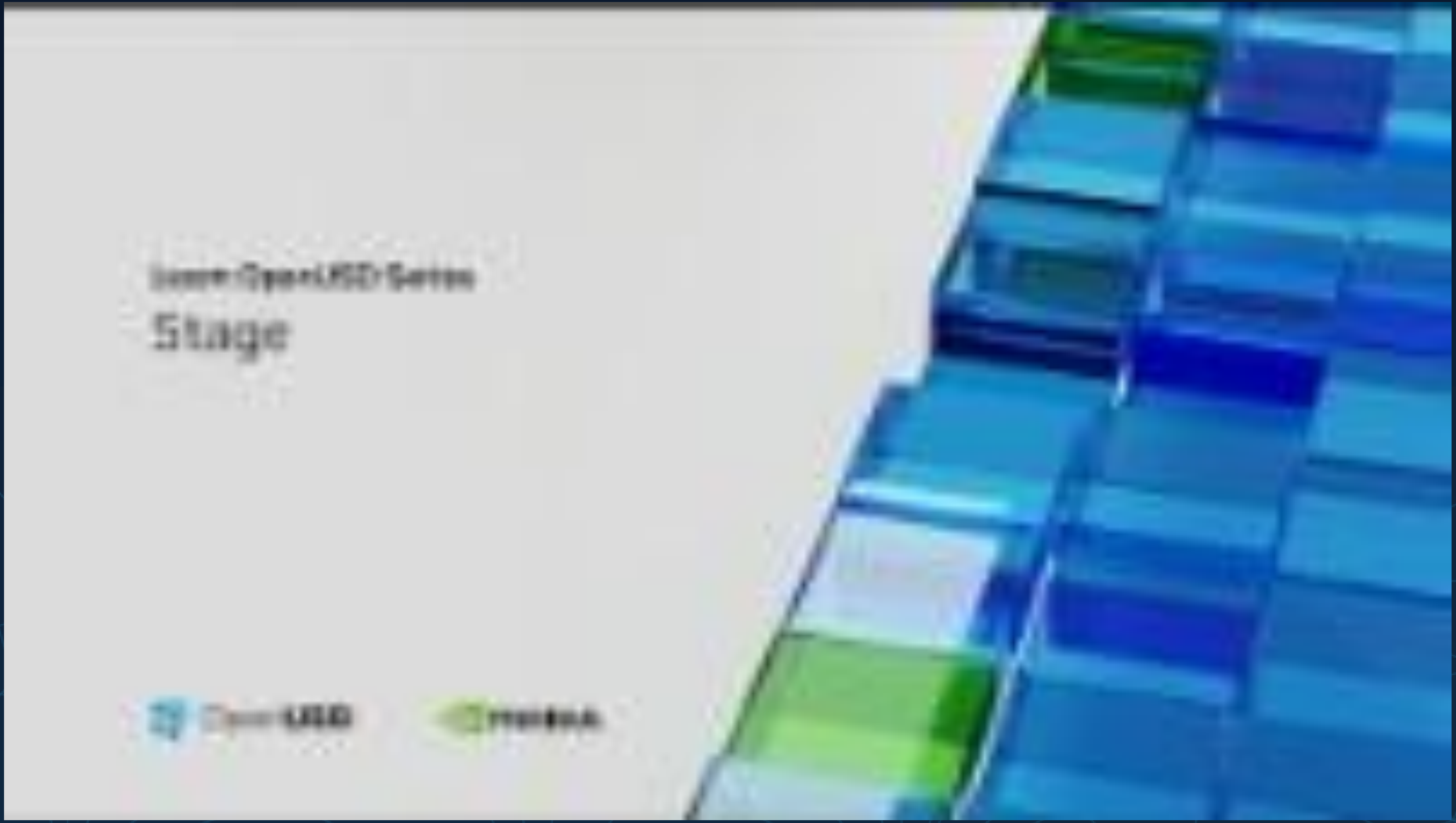
Manual Modelling Phases
Robot Simulation

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:

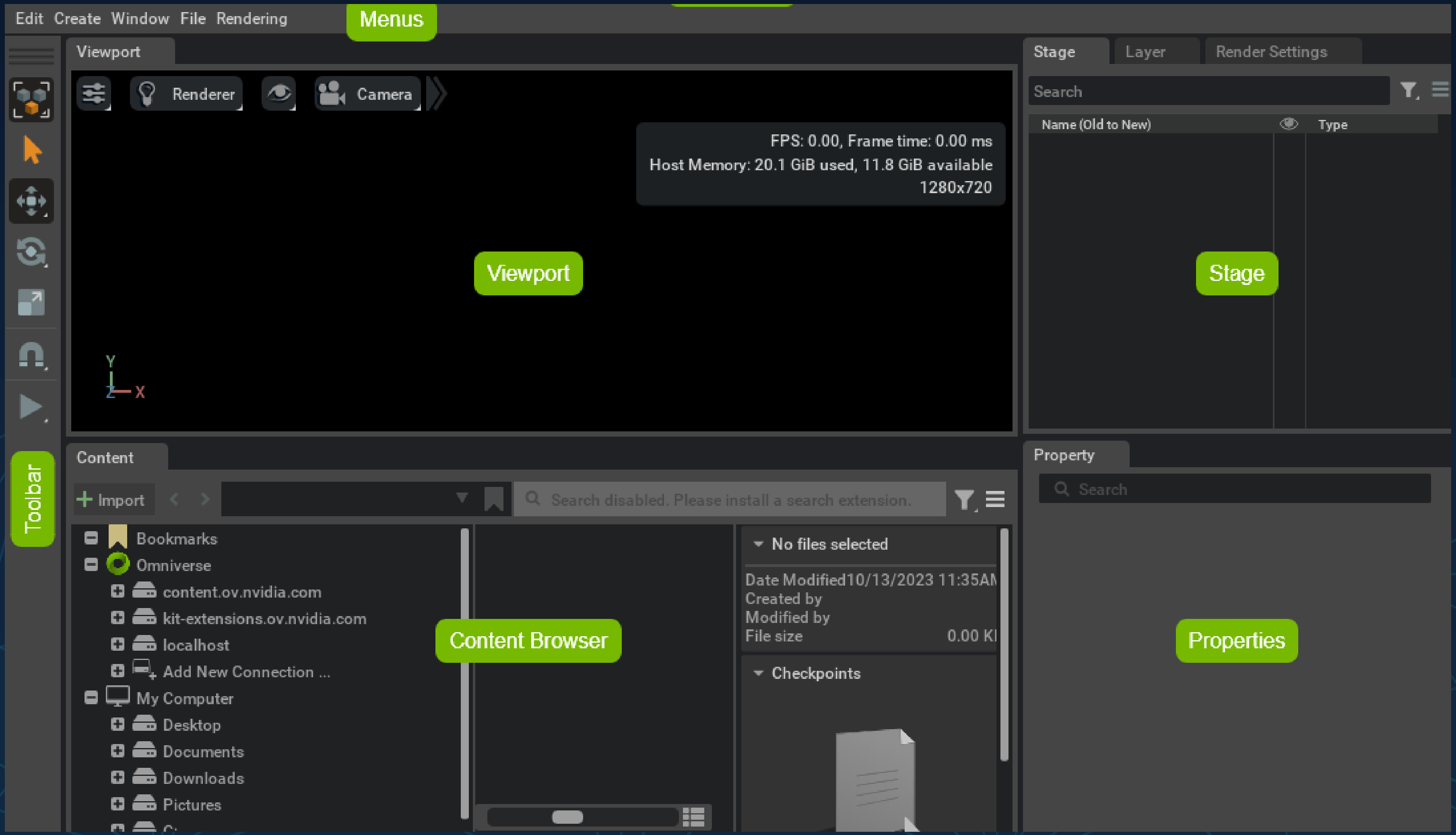
- Practical session:
  - Extensions installation
  - Scene Manipulation
  - Robot import + tuning

Hands on!

Getting started:

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

Standard Layout

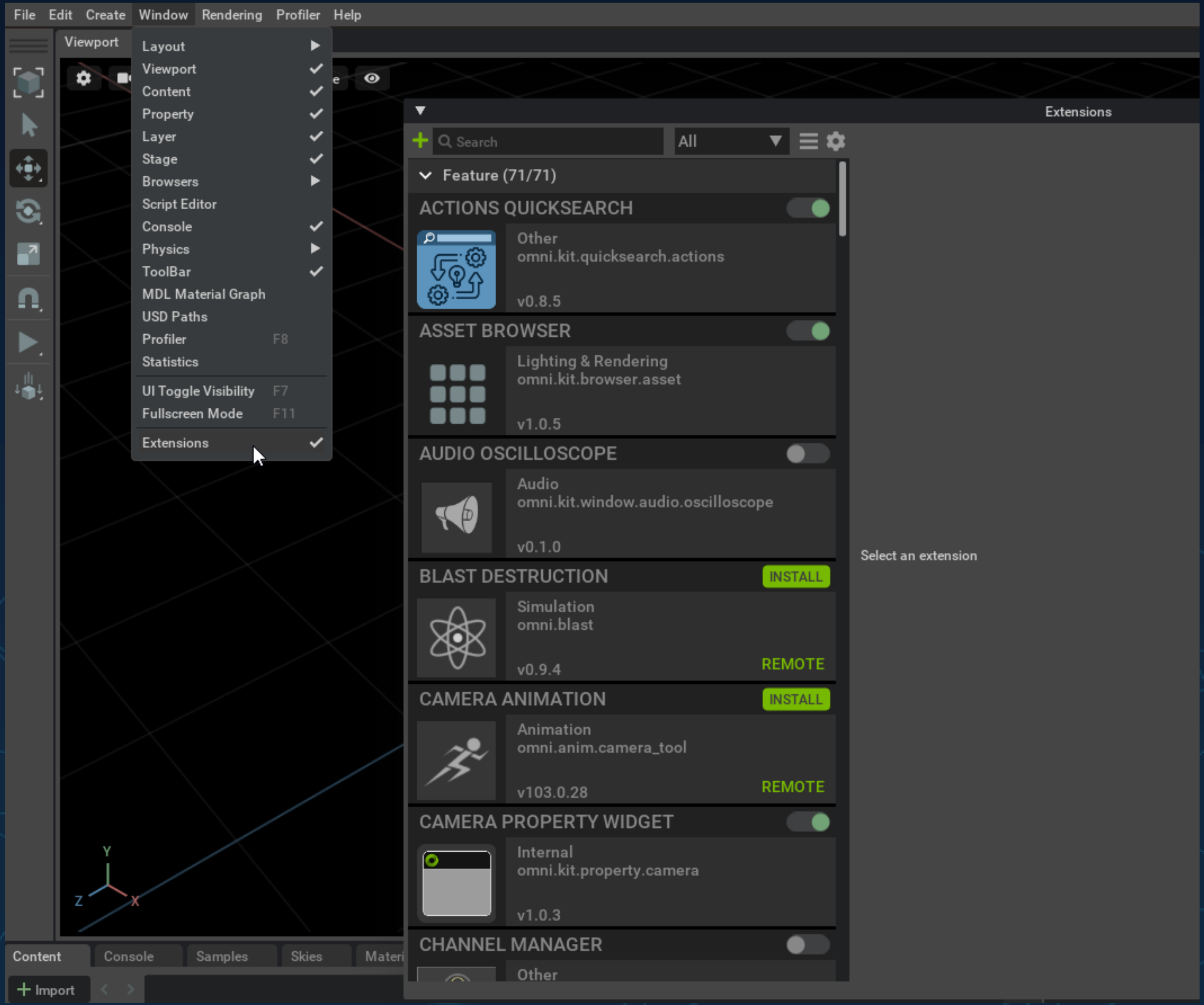


How-to: Viewport Navigation

• 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

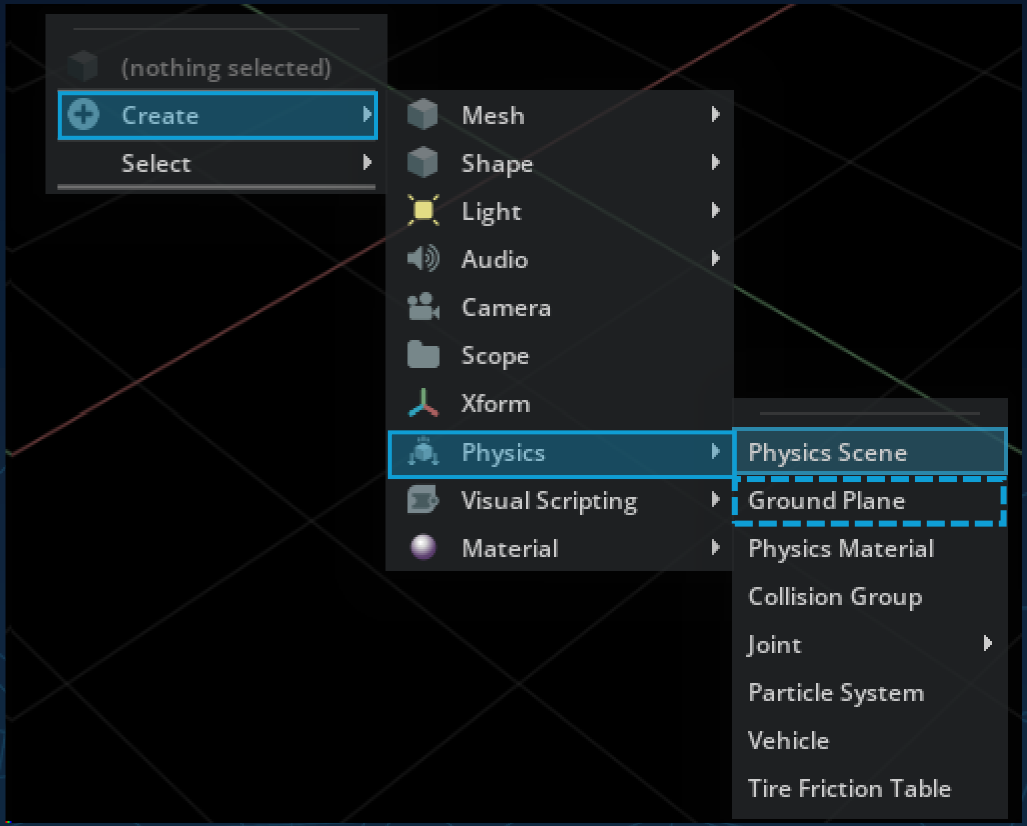
Scene Prep tools (extensions)

- Check for extensions:
  - Action Graph
  - URDF importer



How-to: Add Physics Scene

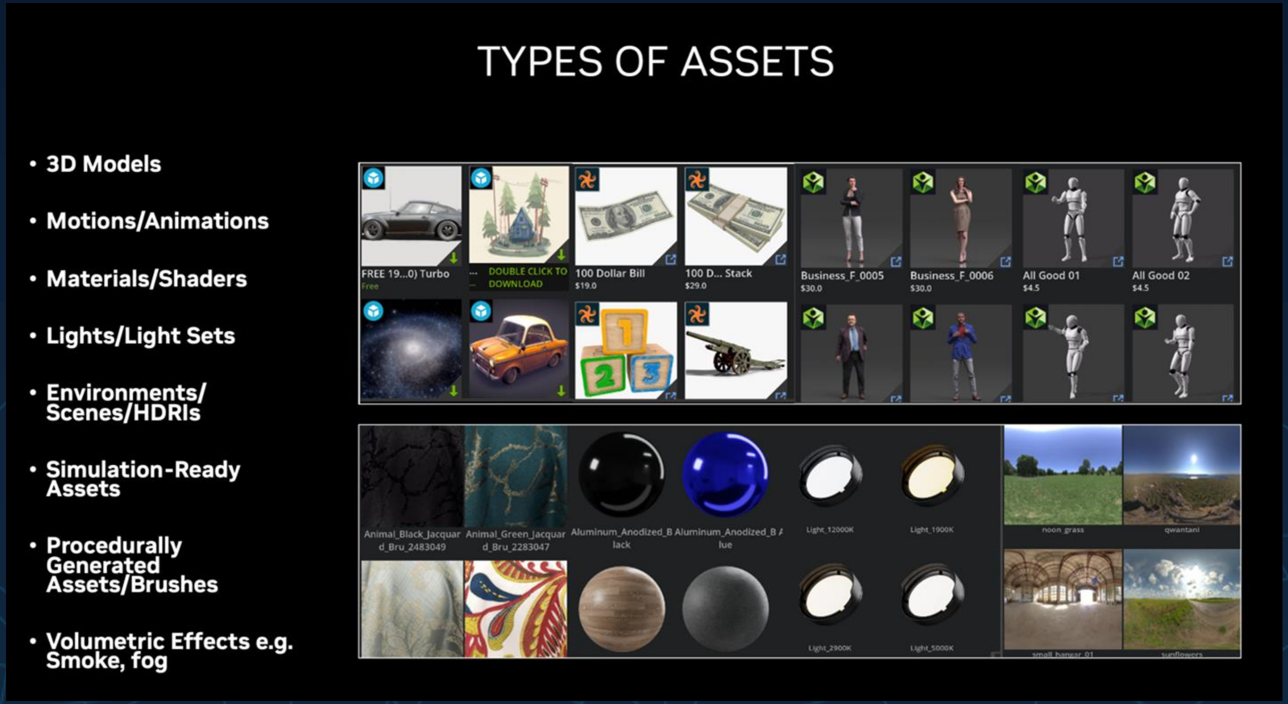
- Create a Physics Scene
- Create a ground plane



IPI
Institut für Produktion und Informatik

Asset Store

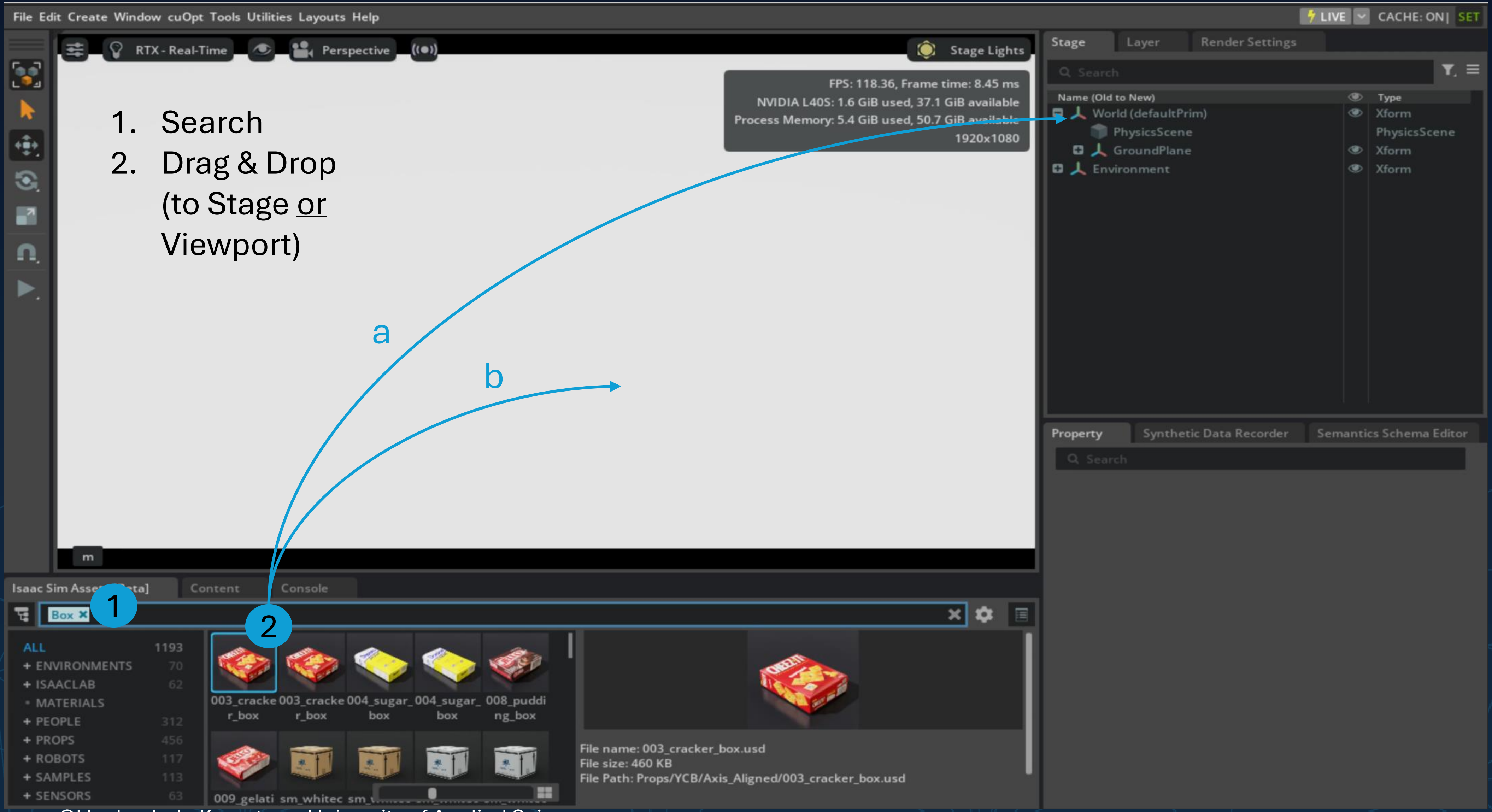
- Import a Box
  - Move above ground plane
  - Adjust the size
  - Add (Rigid Body and and Collider)
  - Press *Play* to start the Simulation



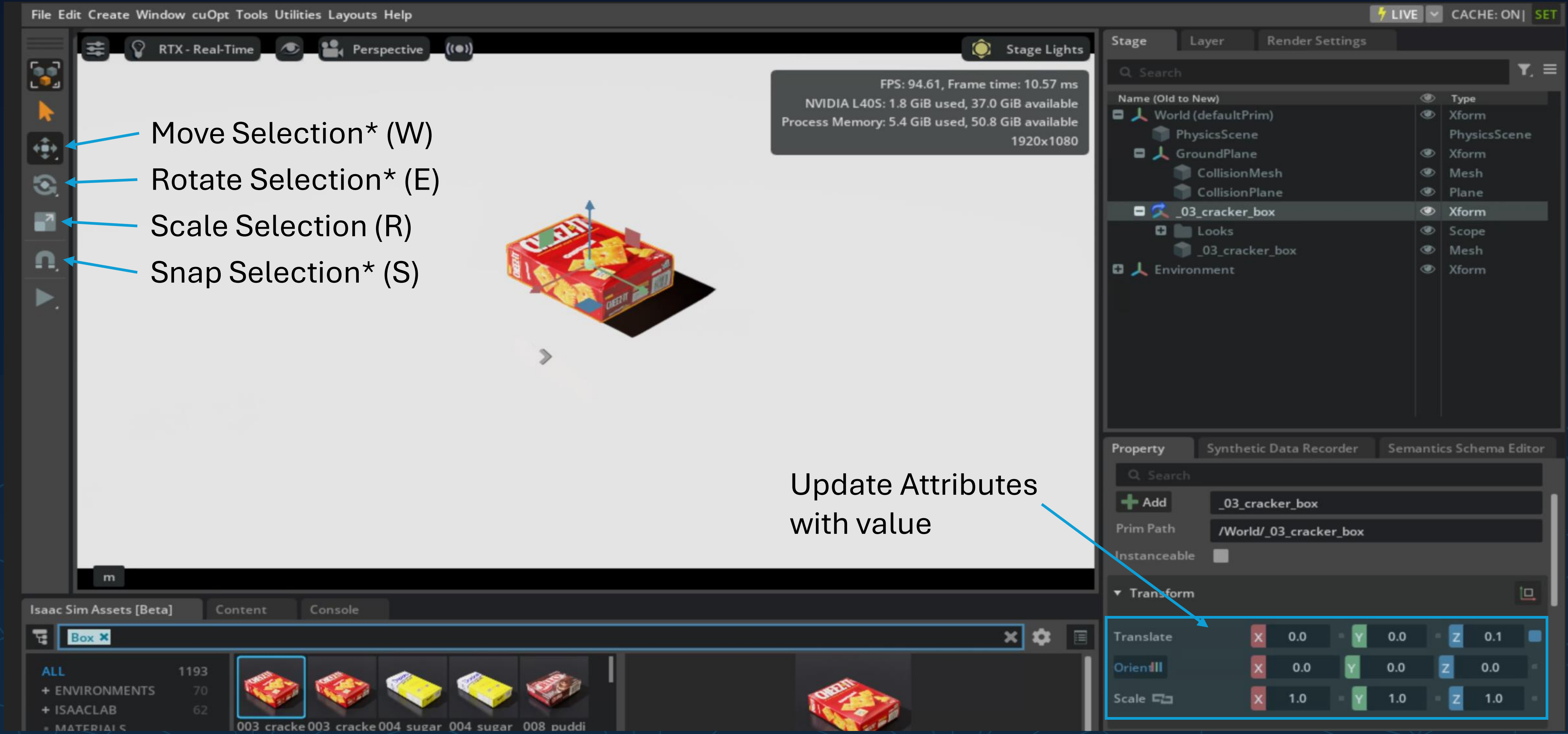
How-to↓

How-to: Use Asset Store

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

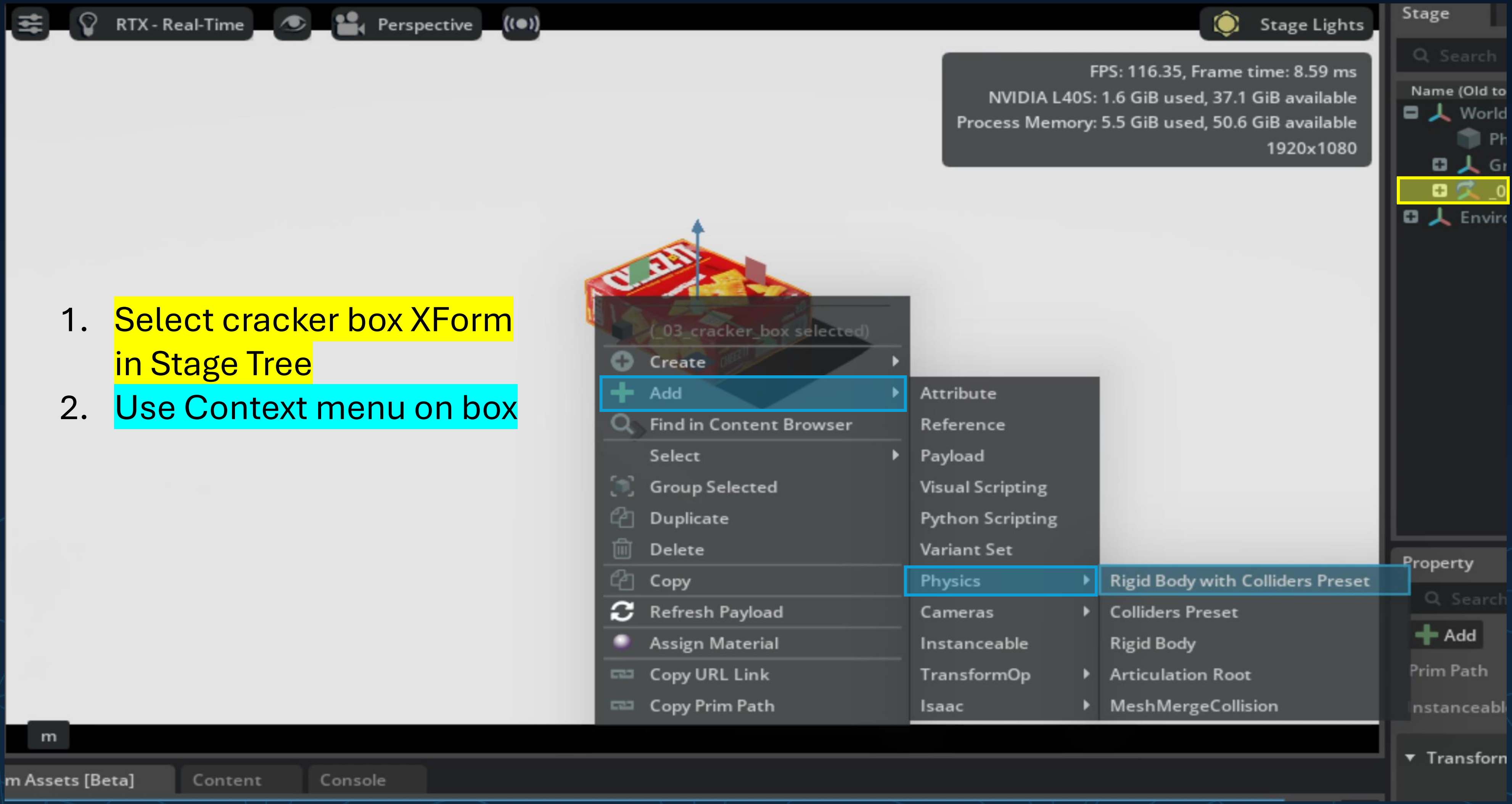


How-to: Manipulation in Viewport



*RMB on icon shows further options

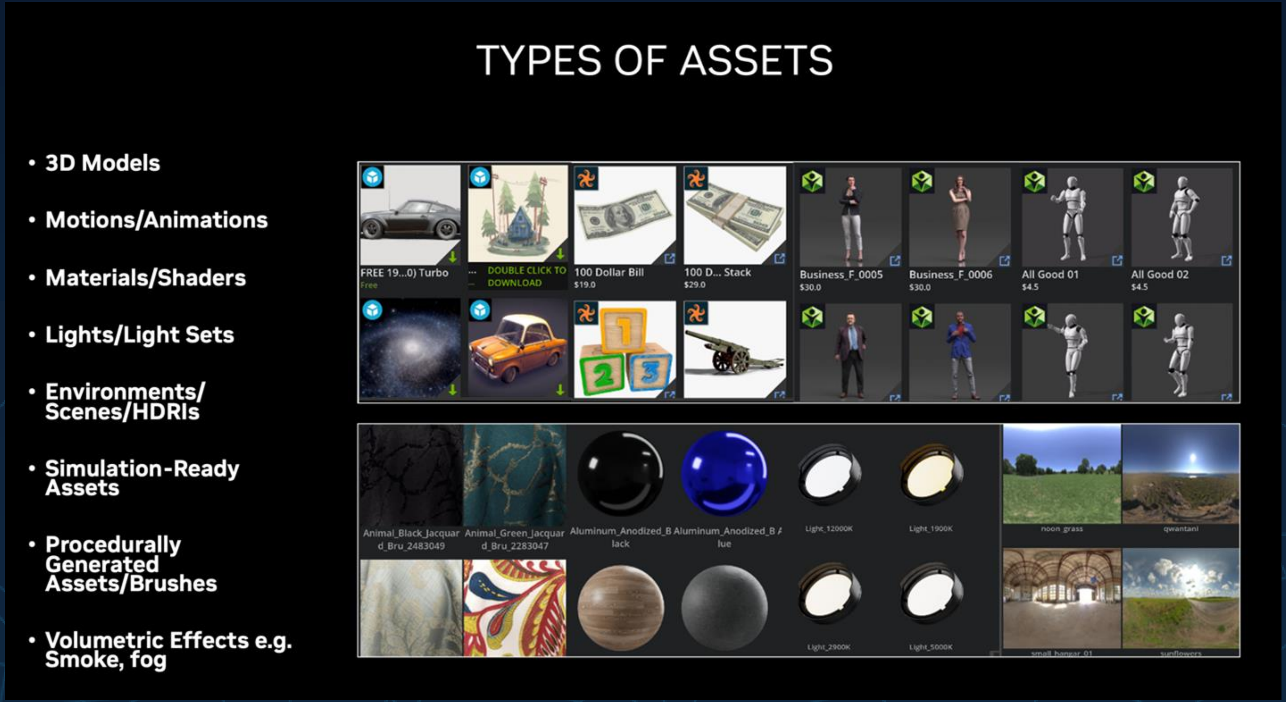
How-to: Add Rigid-body & Collider



1. Select cracker box XForm in Stage Tree
2. Use Context menu on box

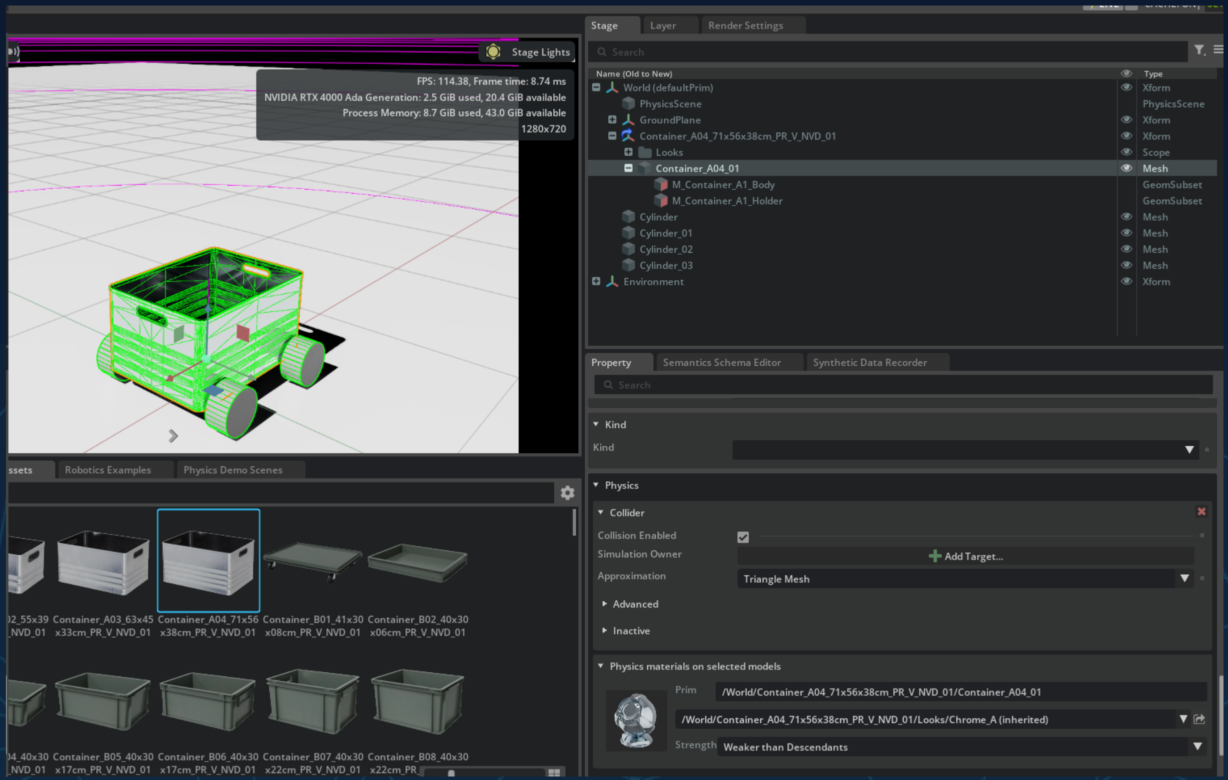
Asset Store

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



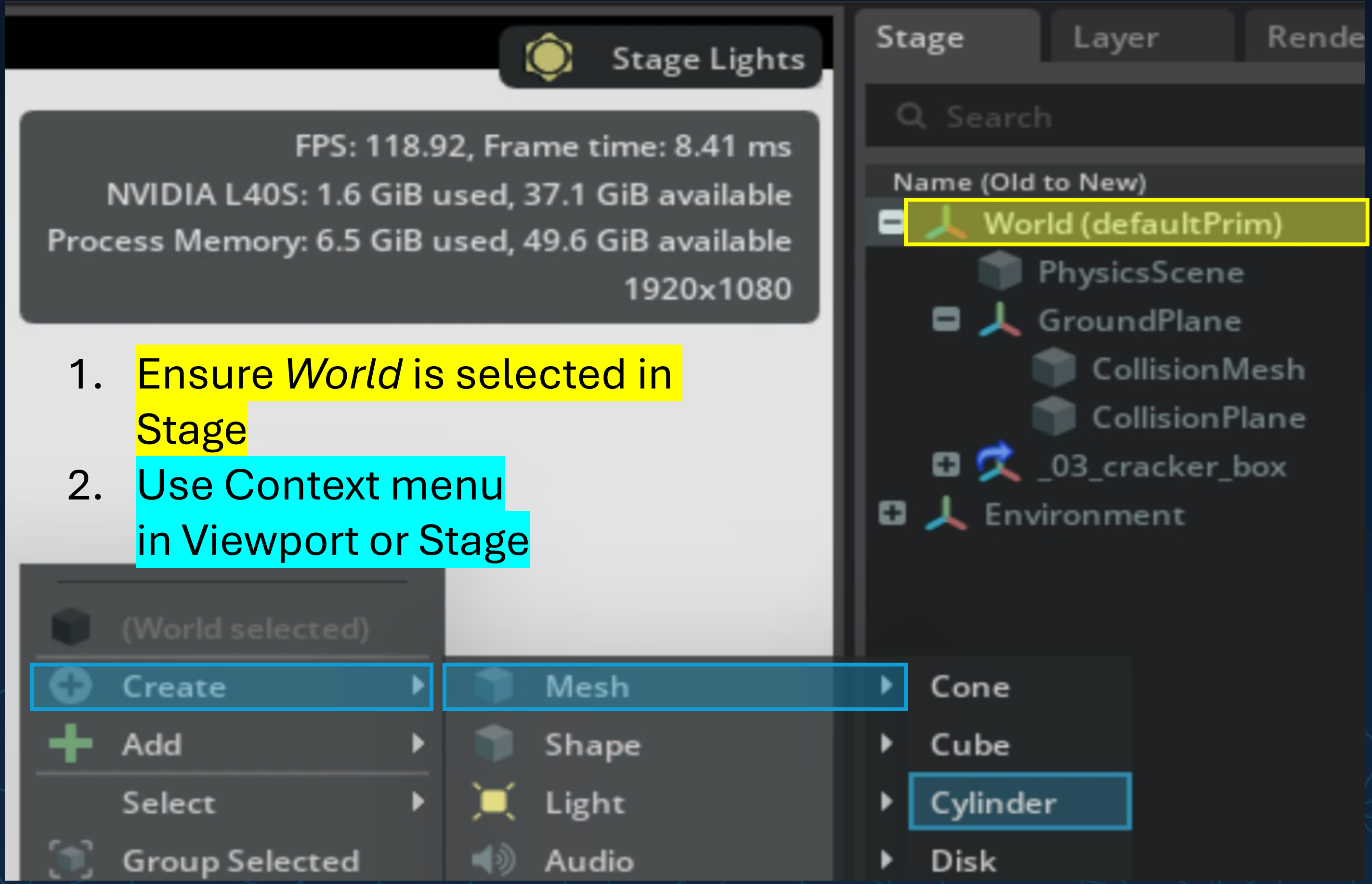
Assemble the first self-built Robot

- 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↓

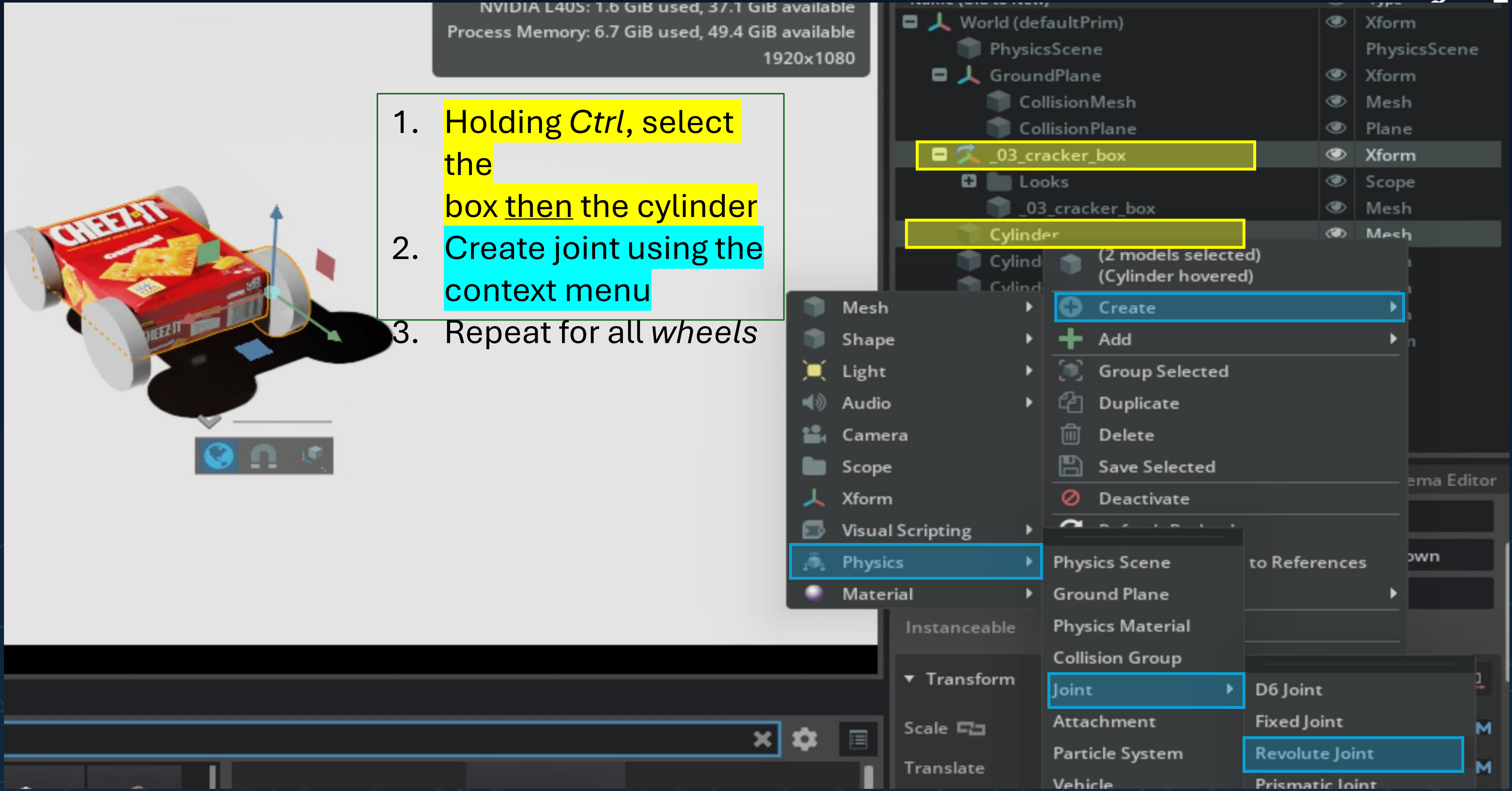
How-to: Add Cylinder



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

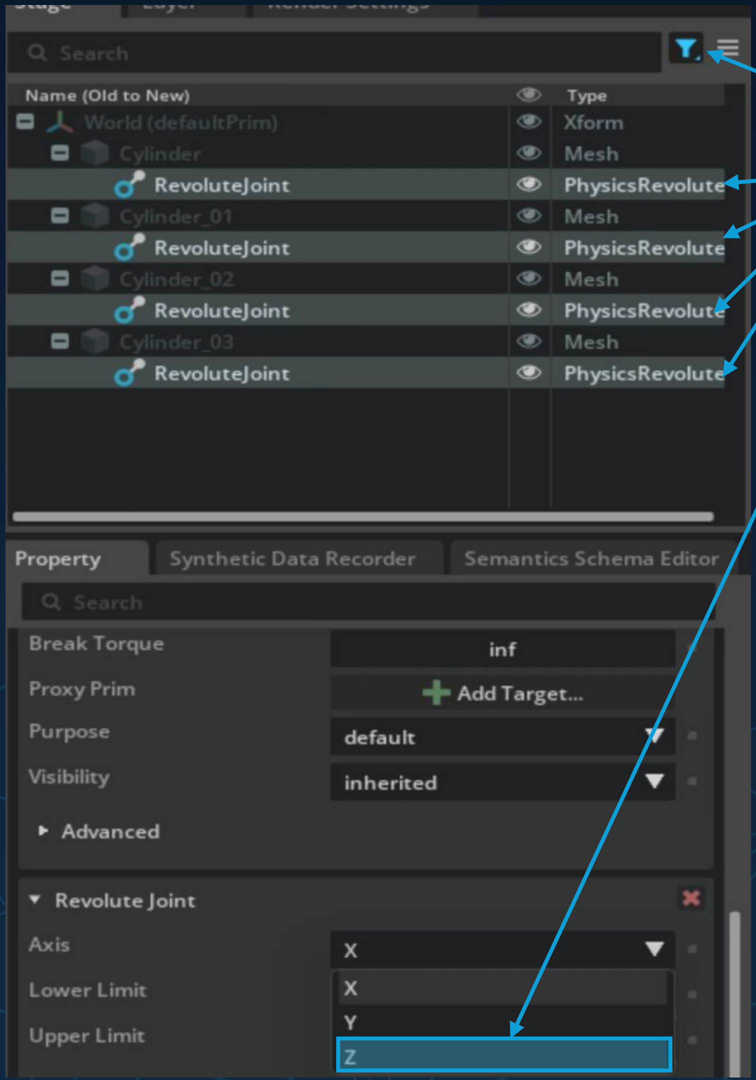
How-to: Add Joint

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

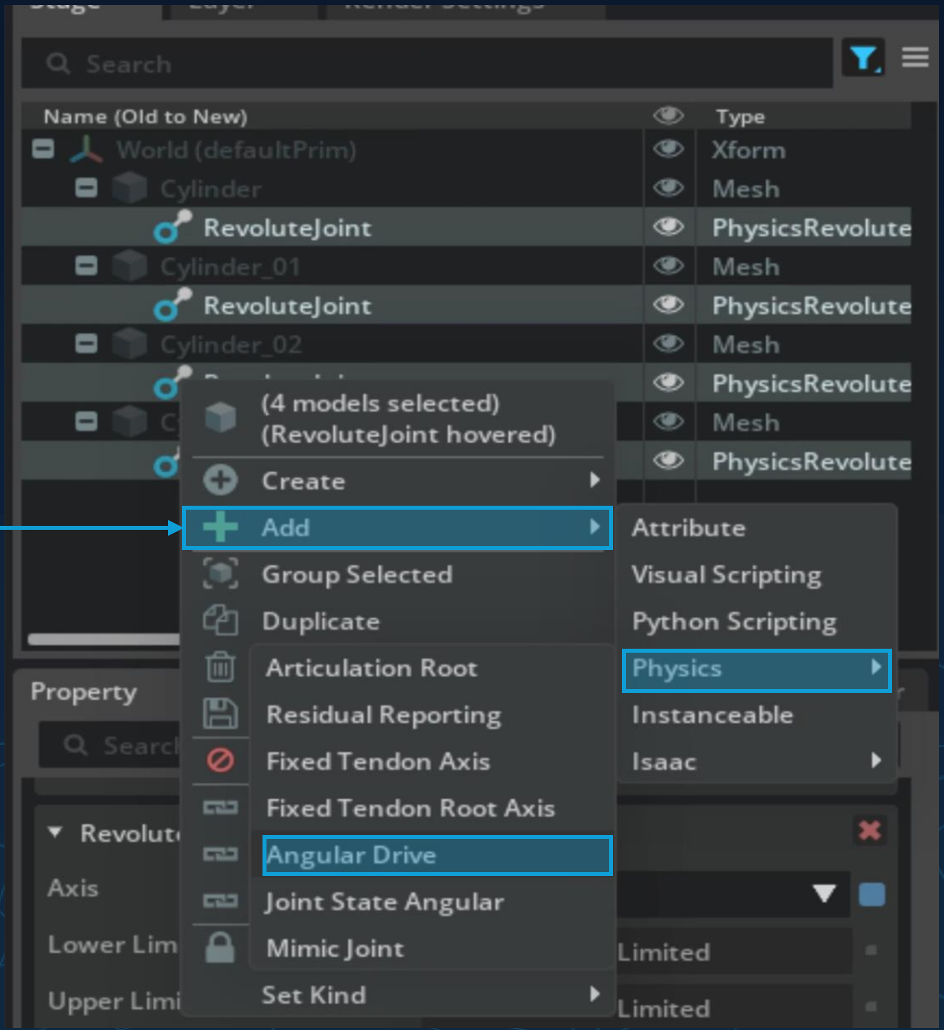


How-to: Adjust Joint and Add Drive

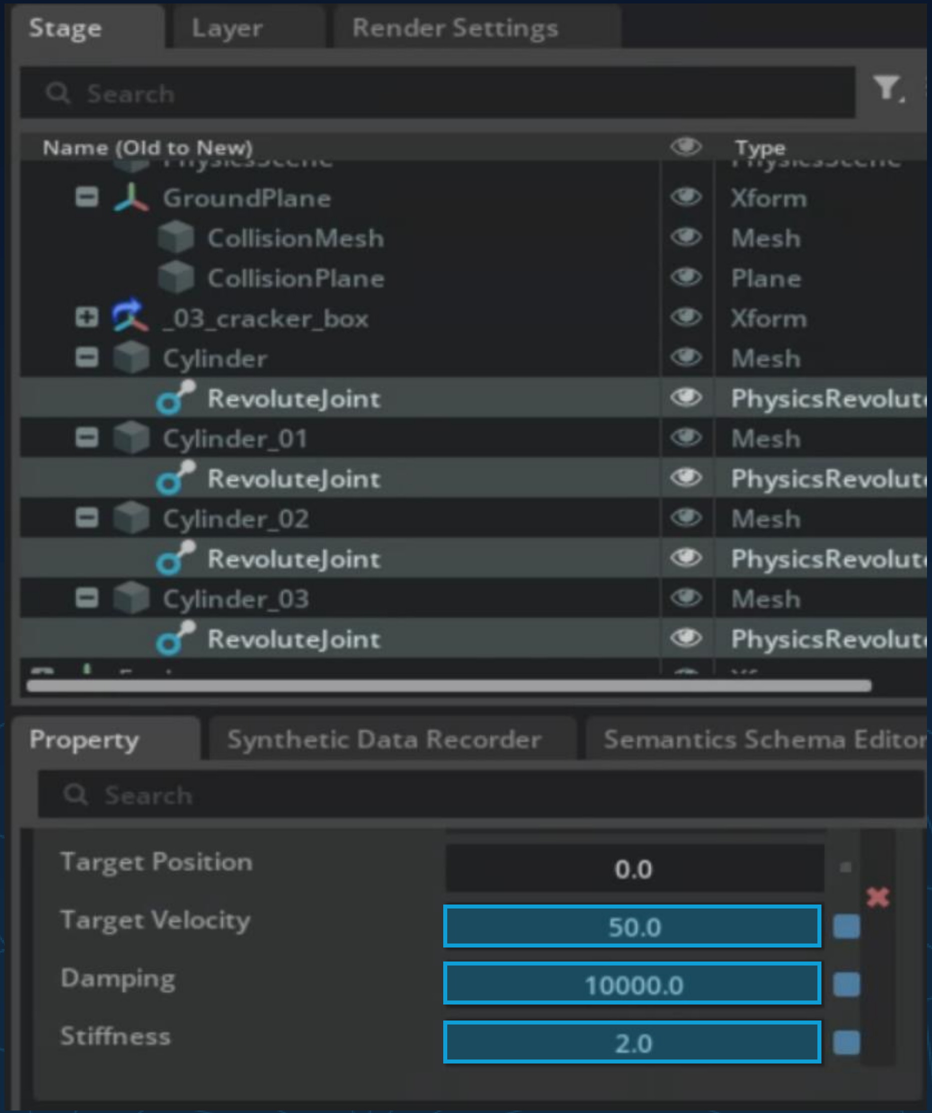
IPI
Institut für Produktion und Informatik



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

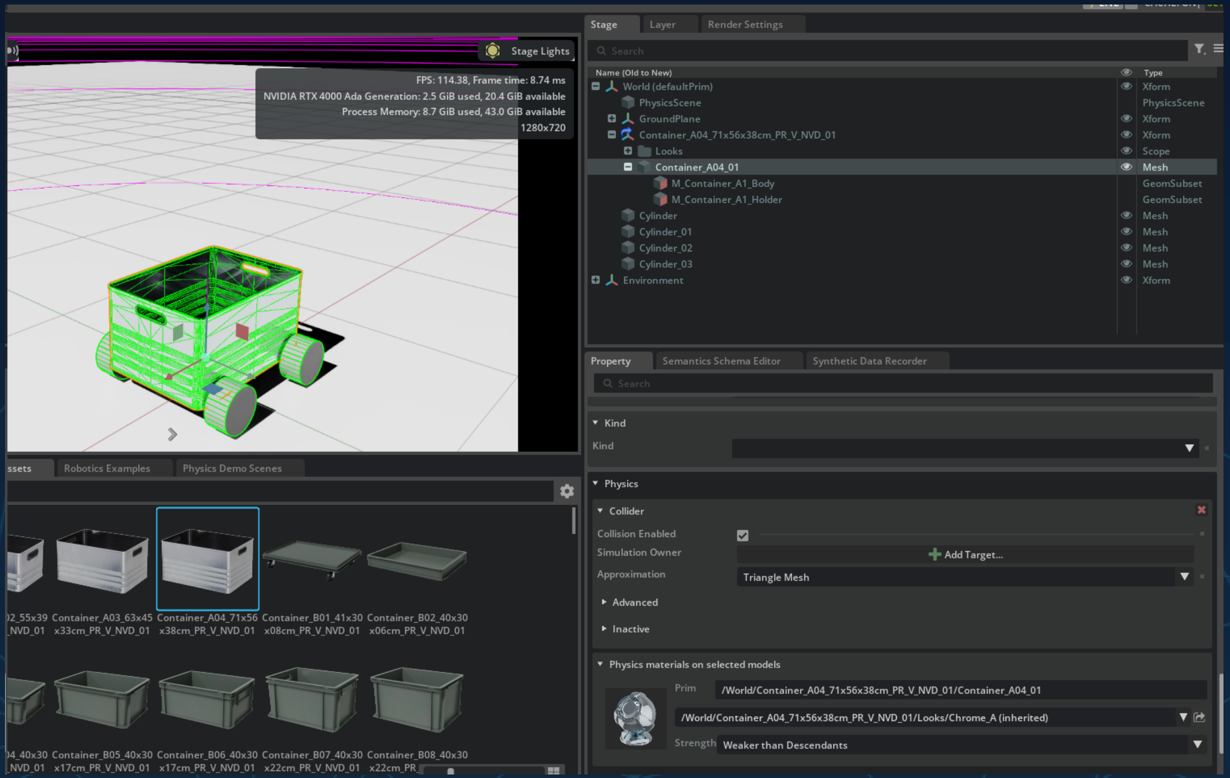


1. Set a Velocity(~20-100)
2. Iterate:
   1. Set Damping & Stiffness Values
   2. Restart Simulation for Testing*

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

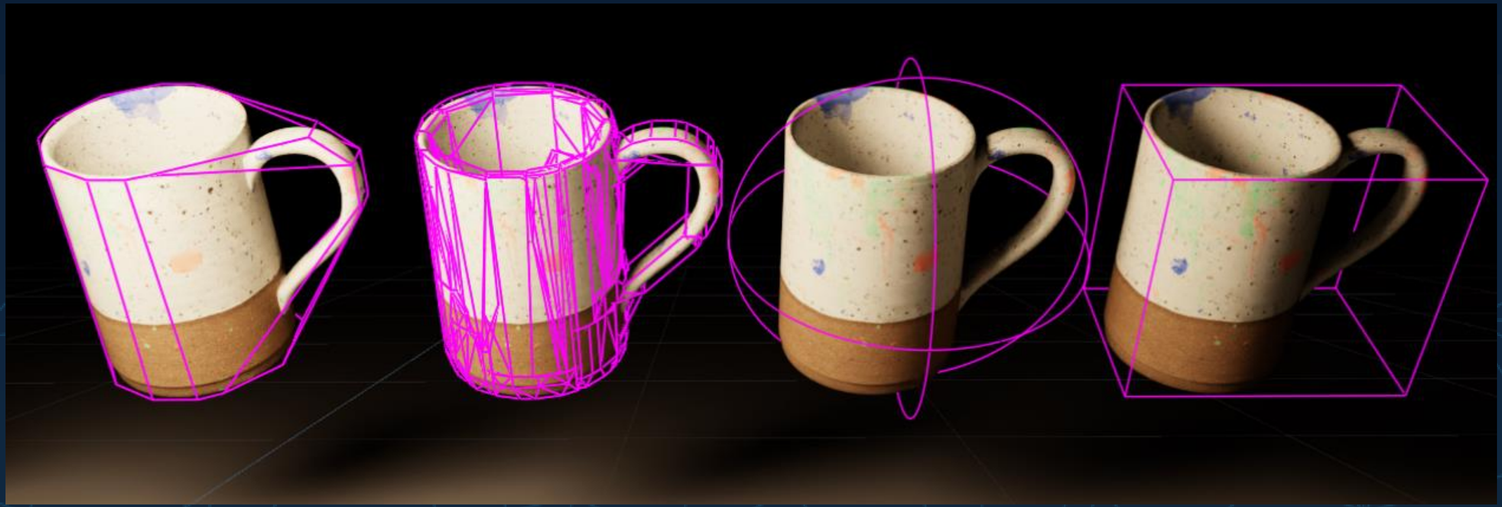
Assemble the first self-built Robot

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

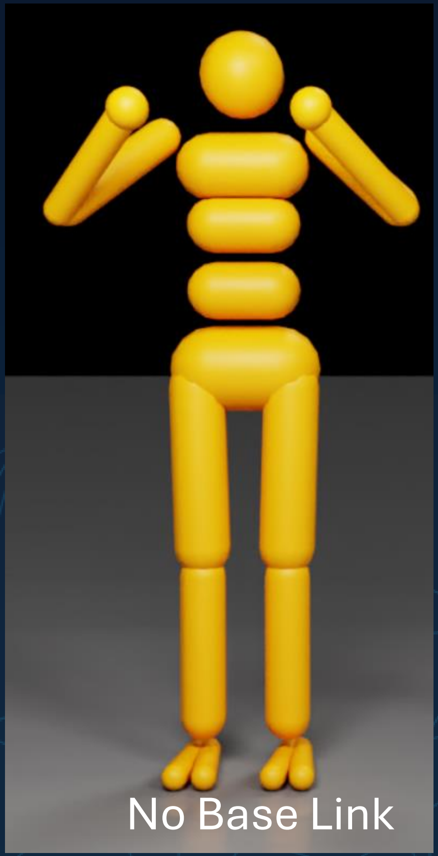


Rigid Bodies and Colliders

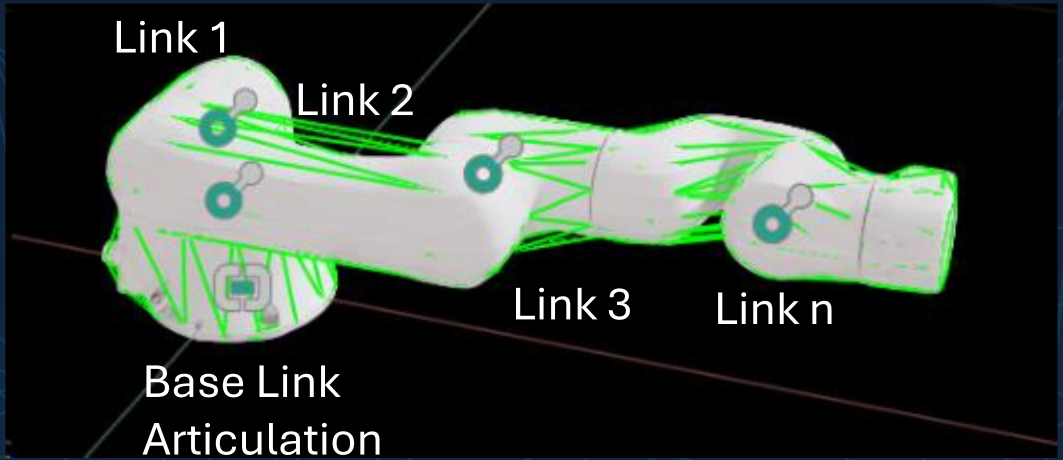
- Simpler approximation = faster Simulation



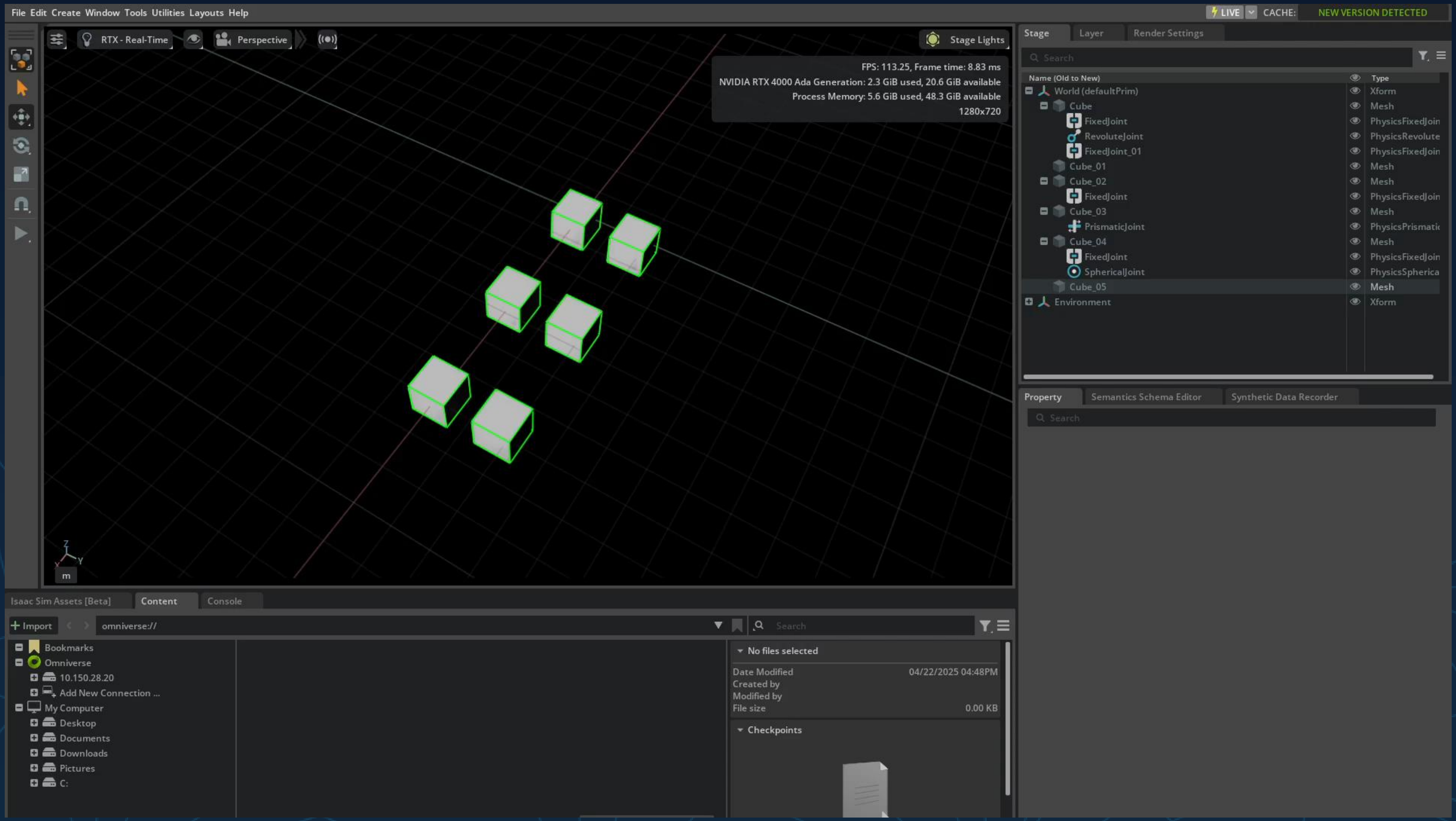
Robot assembly



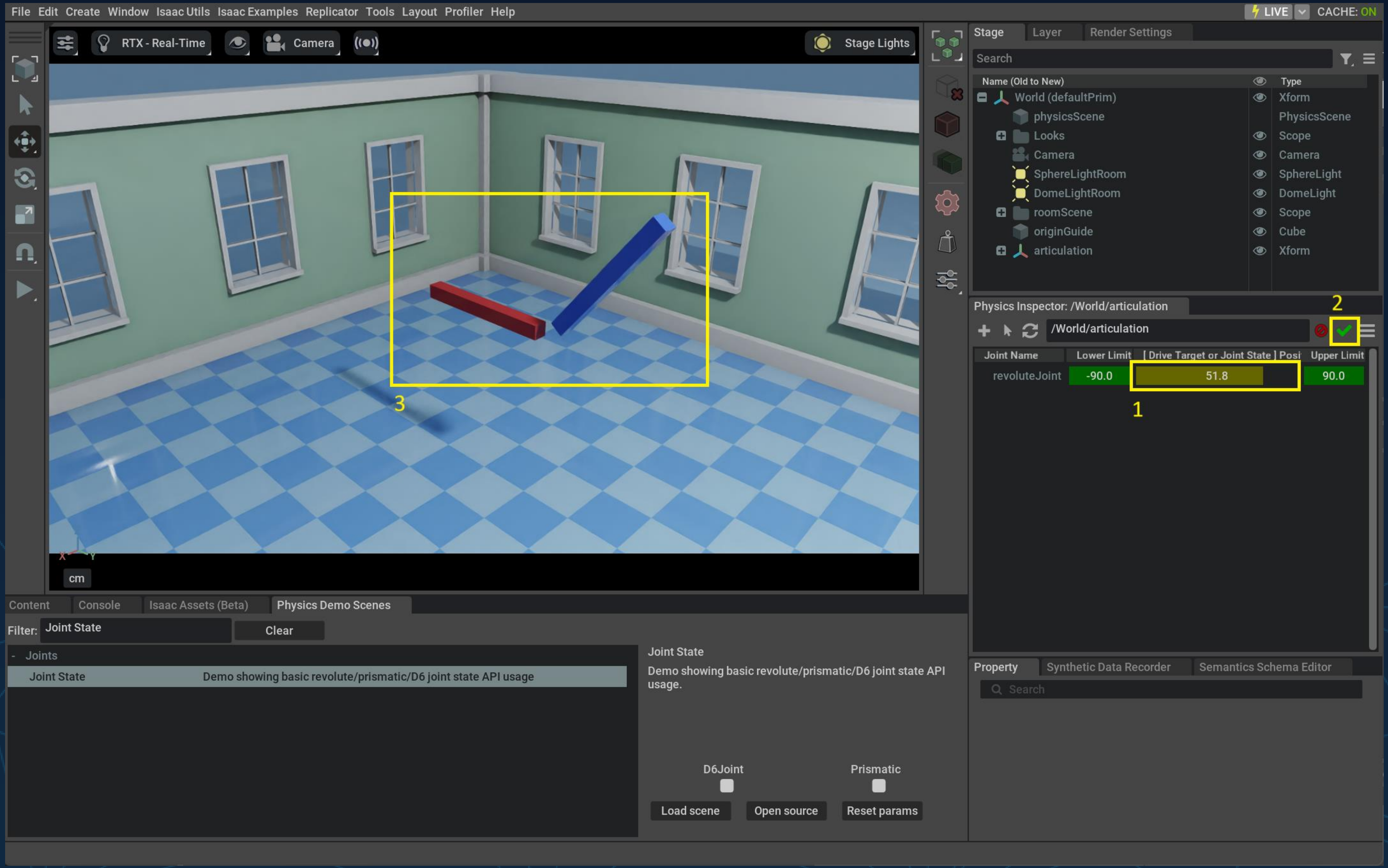
- Structure of mobile / static robot
  - Articulation chain
  - root link free / fixed



Joint types and properties

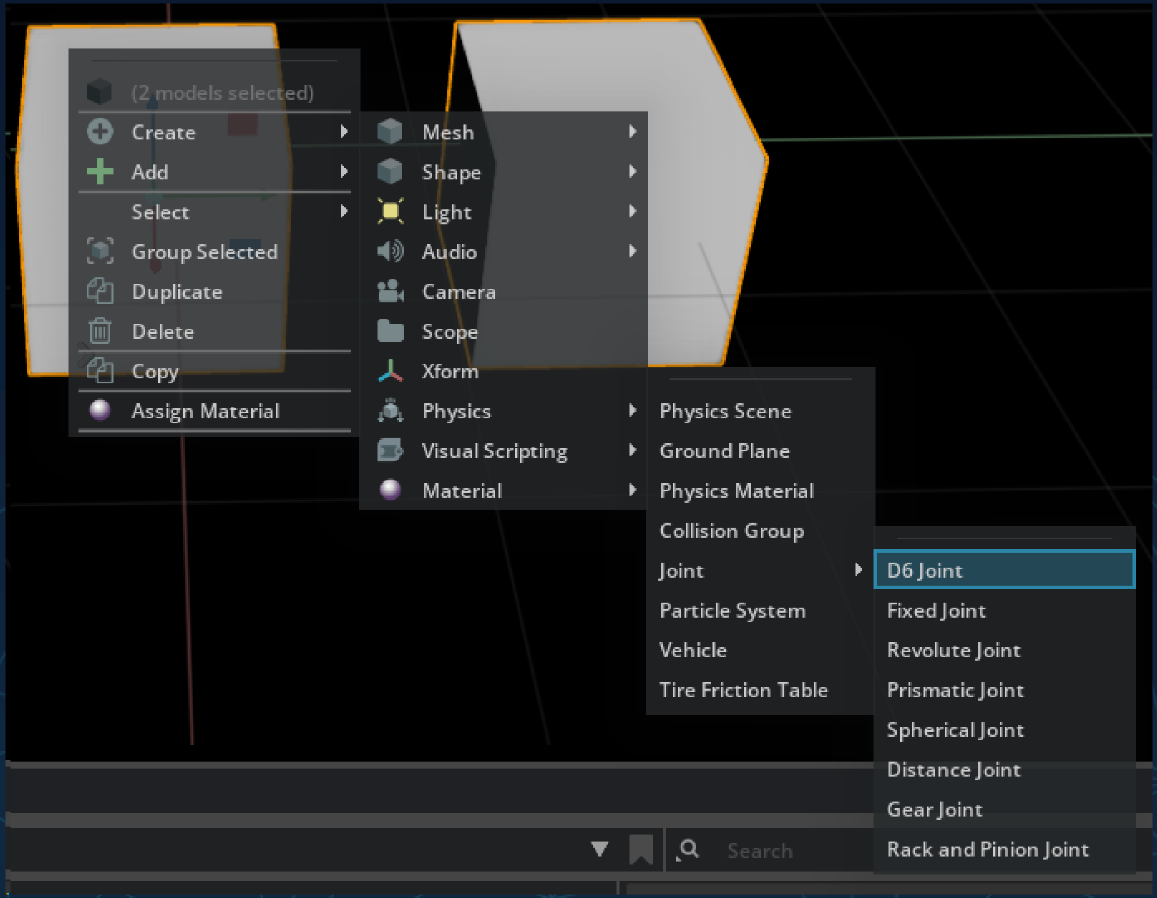


Joints



File Edit Create Window Isaac Utils Isaac Examples Replicator Tools Layout Profiler Help
LIVE
CACHE: ON
RTX - Real-Time
Camera
Stage Lights
Stage
Layer
Render Settings
Search
Name (Old to New)
Type
World (defaultPrim) Xform
physicsScene PhysicsScene
Looks Scope
Camera Camera
SphereLightRoom SphereLight
DomeLightRoom DomeLight
roomScene Scope
originGuide Cube
articulation Xform
Physics Inspector: /World/articulation
/World/articulation
Joint Name
Lower Limit
[Drive Target or Joint State] Posi
Upper Limit
revoluteJoint
-90.0
51.8
90.0
1
2
3
cm
Content
Console
Isaac Assets (Beta)
Physics Demo Scenes
Filter: Joint State
Clear
Joints
Joint State
Demo showing basic revolute/prismatic/D6 joint state API usage
Joint State
Demo showing basic revolute/prismatic/D6 joint state API usage.
D6Joint
Prismatic
Load scene
Open source
Reset params
Property
Synthetic Data Recorder
Semantics Schema Editor
Search

Joint types and properties



Joint drives are internally simulated with PD-Controllers

- *Stiffness* property <-> Proportional coefficient
- *Damping* property <-> 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

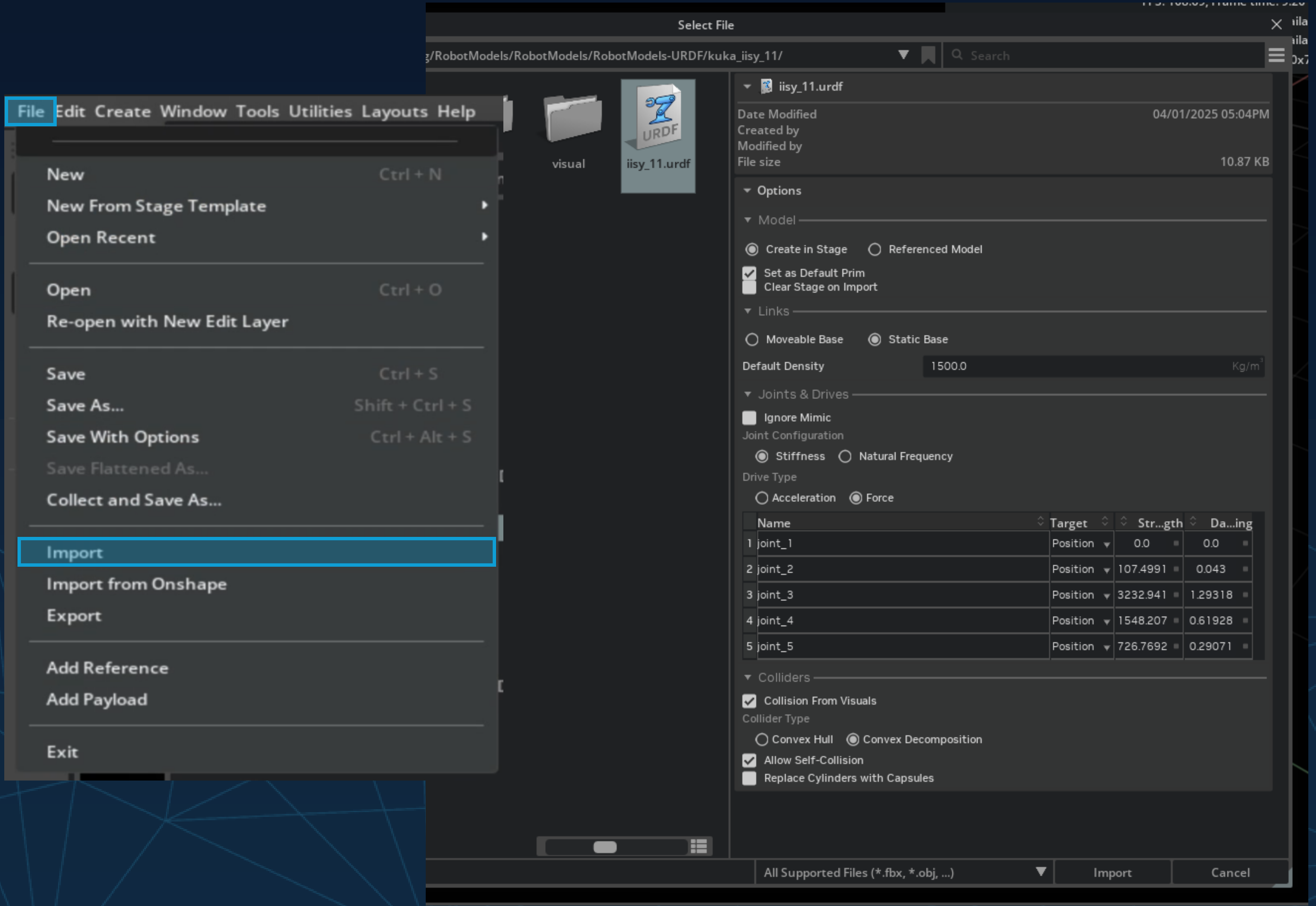
10 min Break

To be continued:

- Practical session:
  - Robot import + tuning

How-to: Starting with a real Robot

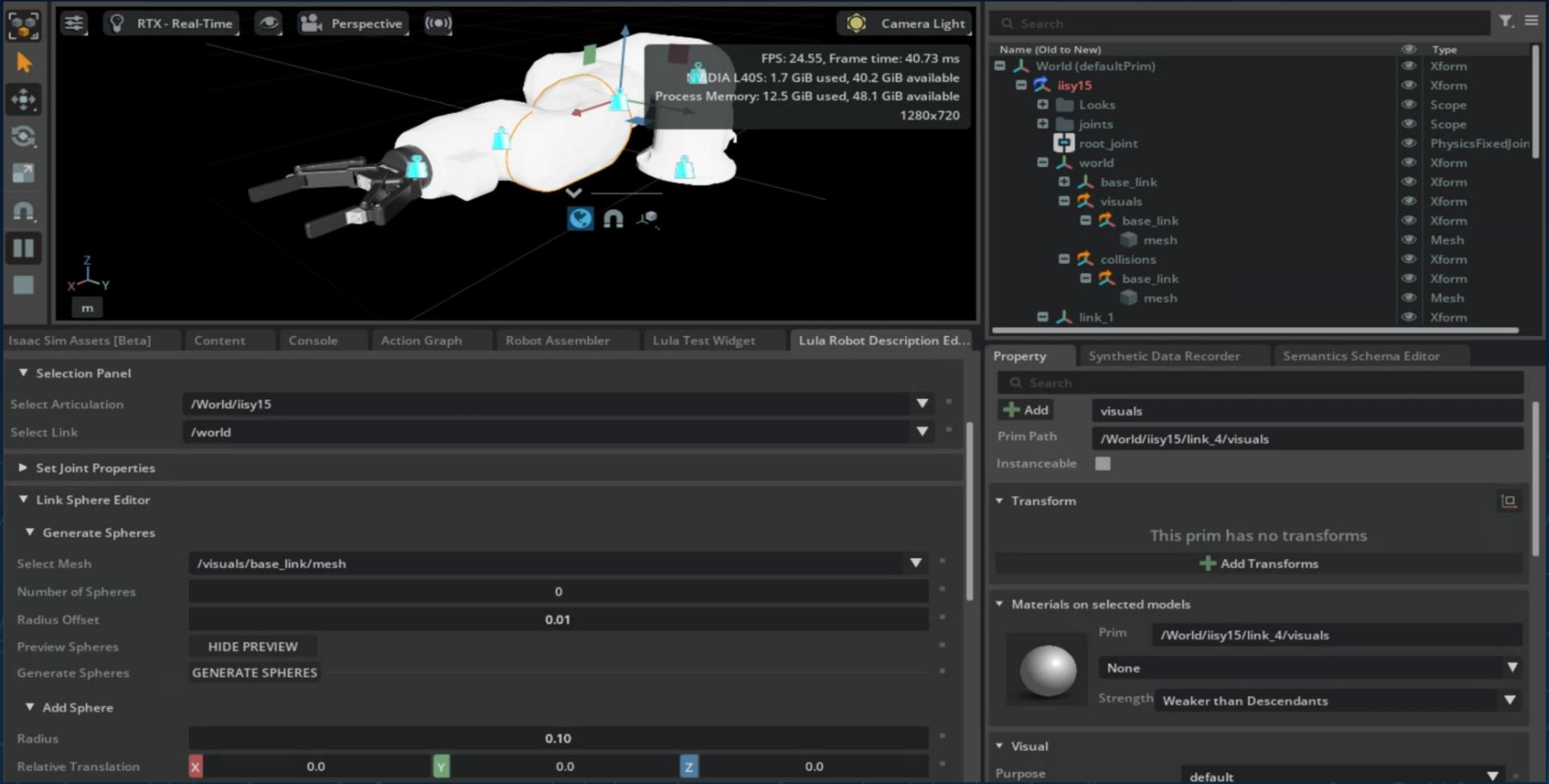
- Open *File->import*
- Navigate to a URDF File
- Modify the Options
  - Verify *Model* and *Links*
  - Adjust *Joints & Drives*
  - Verify *Colliders*
- Click *Import*



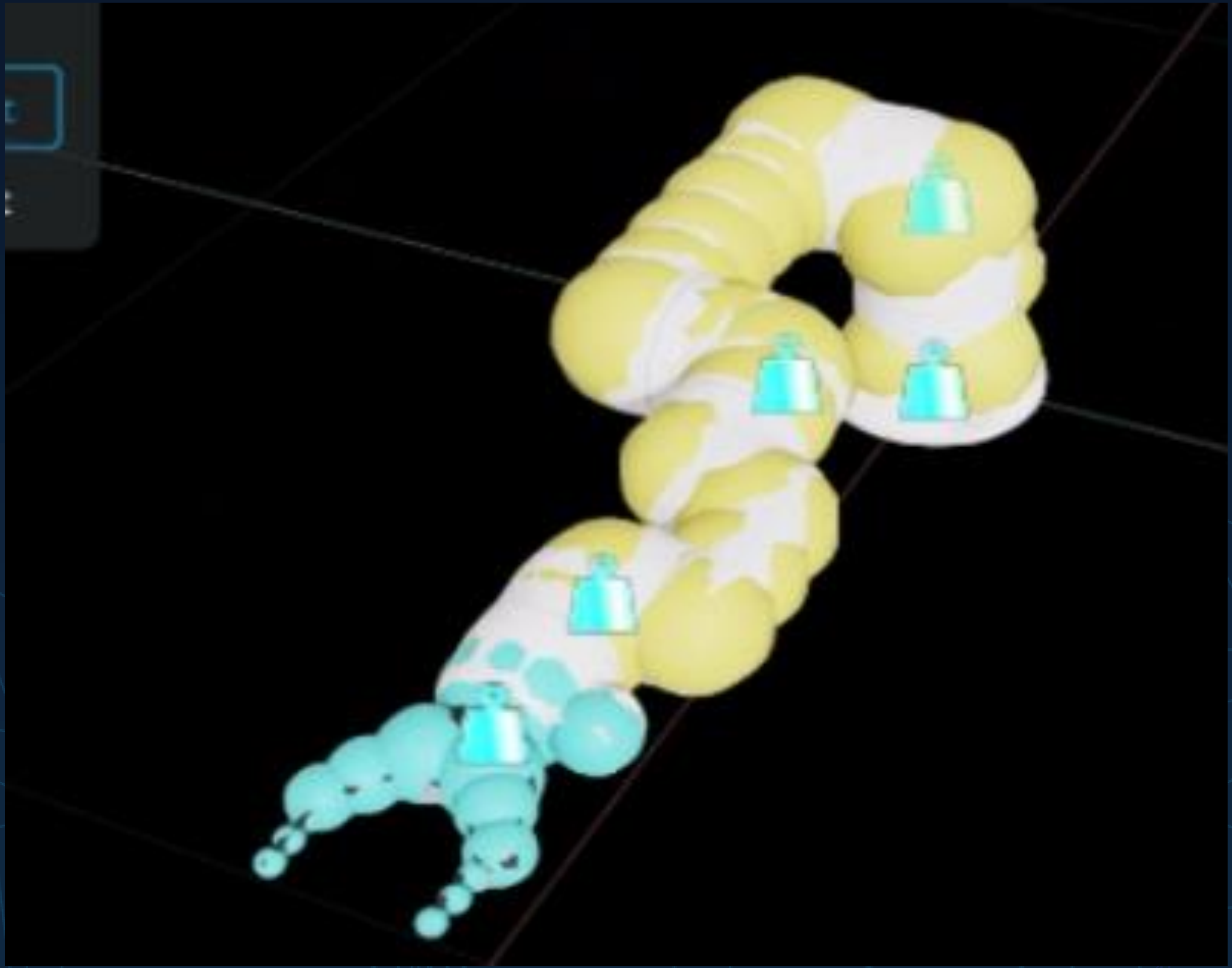
Robot import Tools

- 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

Robot Description Editor

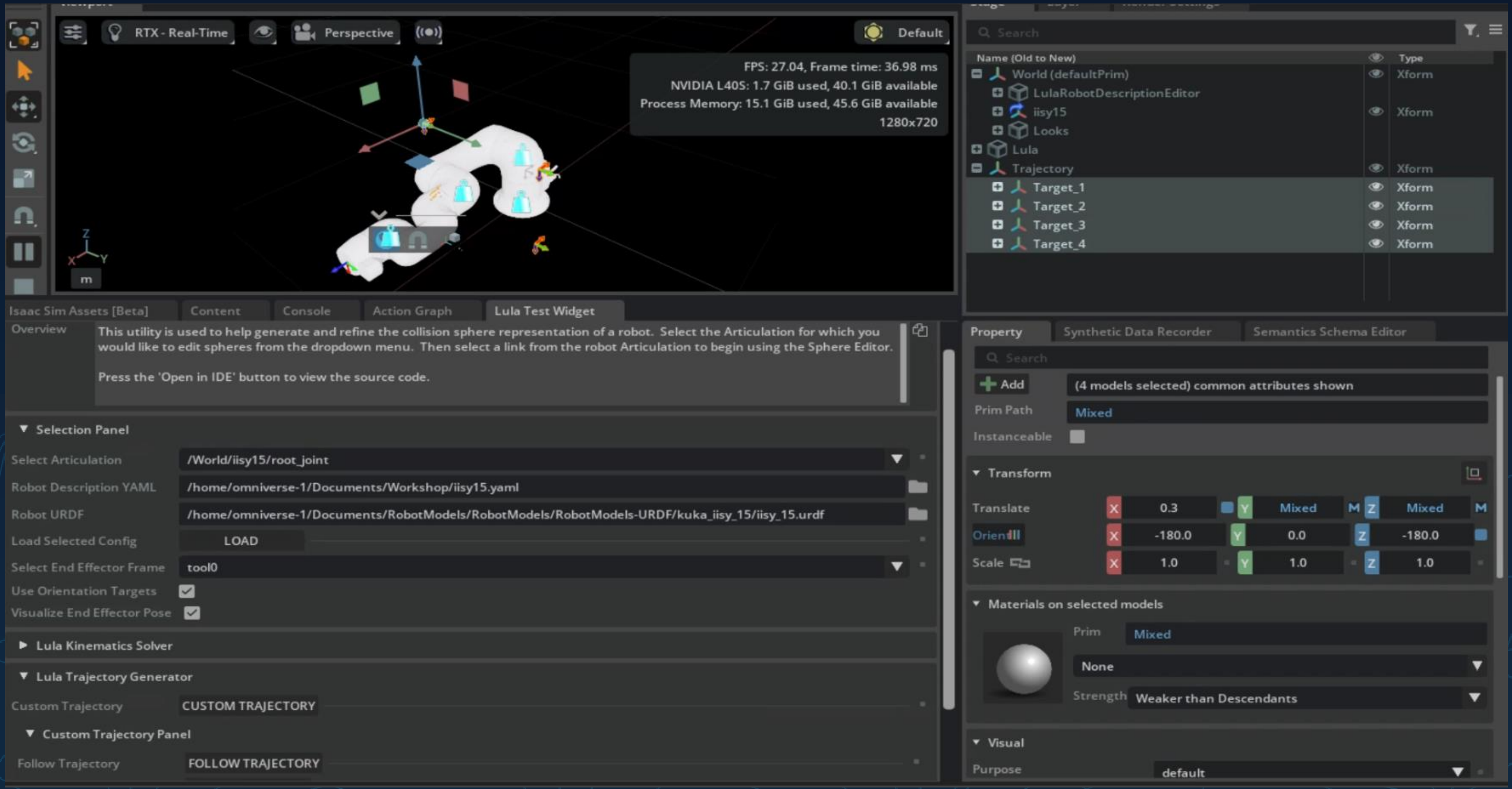


Sphere Approximation Result

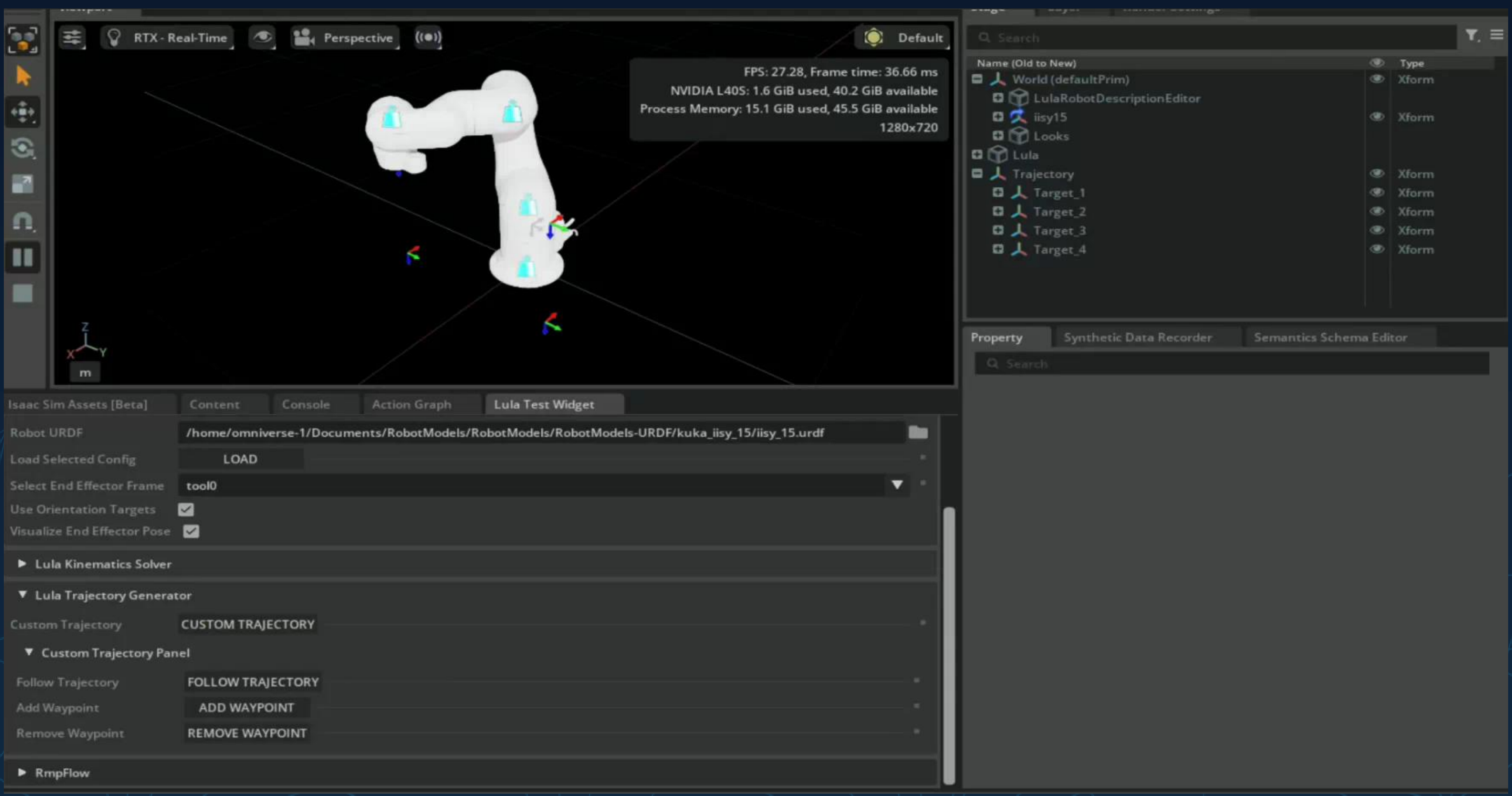


Lula Test Widget

IPI Institut für Produktion und Informatik



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