

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



Institute for Production and Informatics

Technology-transfer-center from the university of applied sciences Kempten

- Application-oriented research facility
- Research and development projects surrounding industry 4.0
- Founded 2021 with now 30 employees
- Technologytransfer and startup sncubator
- Virtual Comissioning and industrial metaverse in education





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



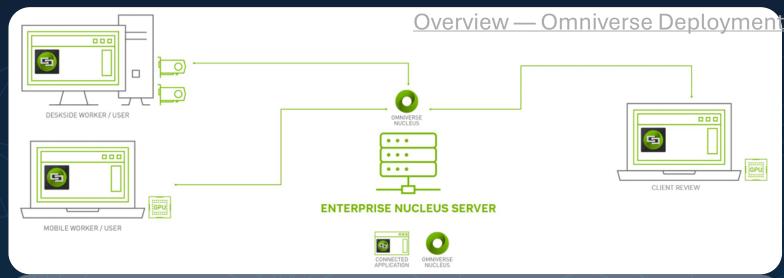
Omniverse Architecture





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





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

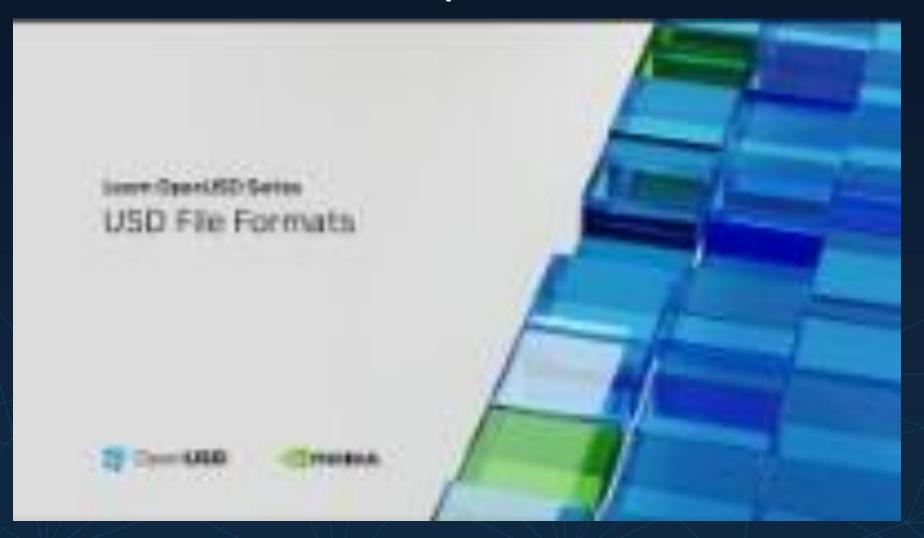
USD

Viewport

Isaac Automator



Universal scene description





Interacting with USD (in Omniverse scene)

- 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	NVIDA CuMotion and Lula to generate Motion Plans





- 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!

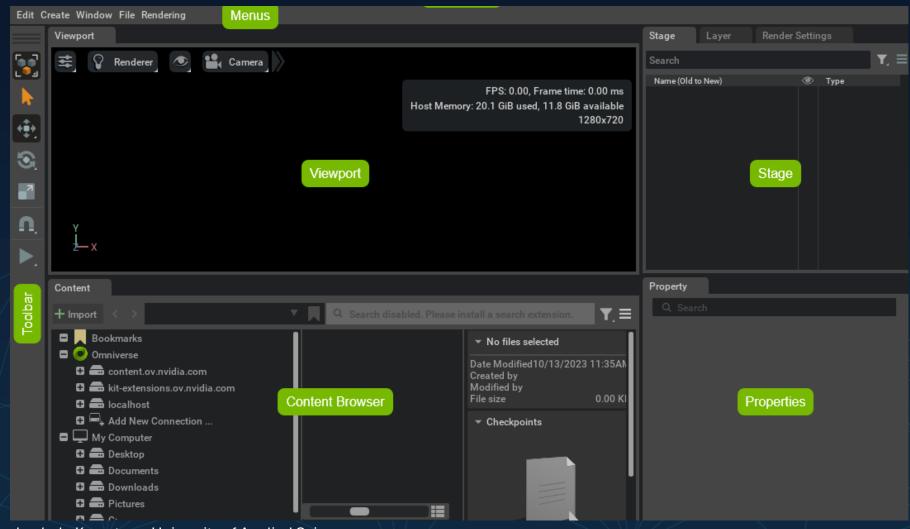


Getting started:

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



Standard Layout



©Hochschule Kempten – University of Applied Sciences



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

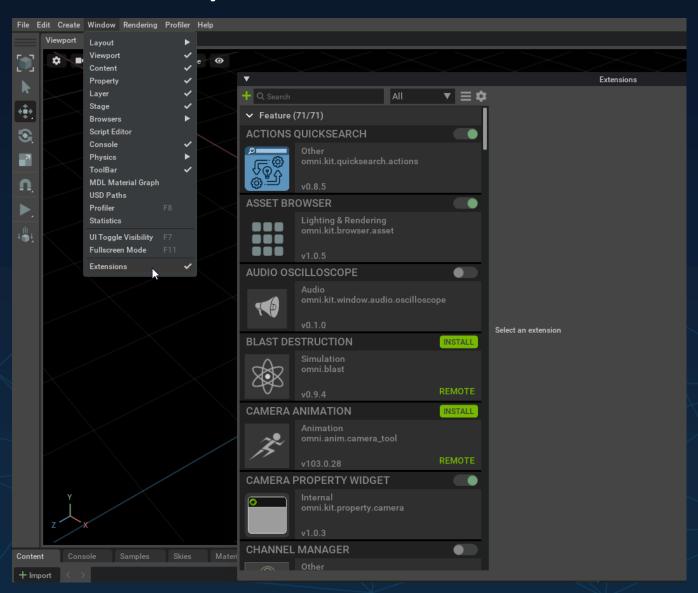
Jump to Selection:





Scene Prep tools (extensions)

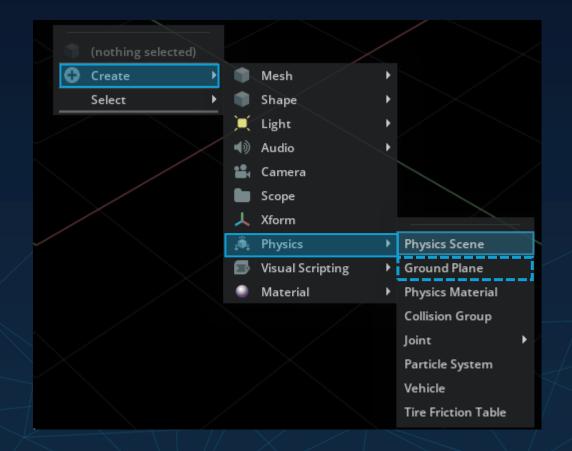
- Check for extensions:
 - Action Graph
 - URDF importer





How-to: Add Physics Scene

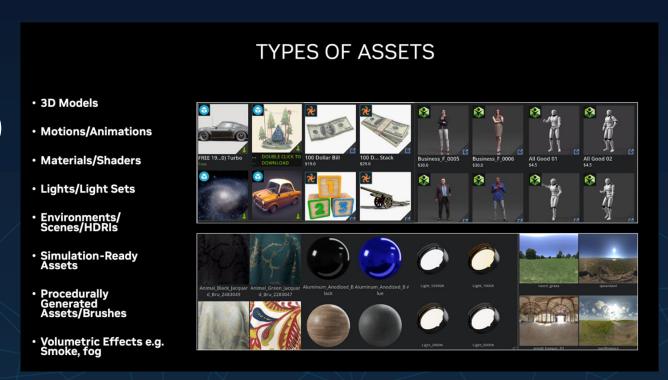
- Create a Physics Scene
- Create a ground plane





Asset Store

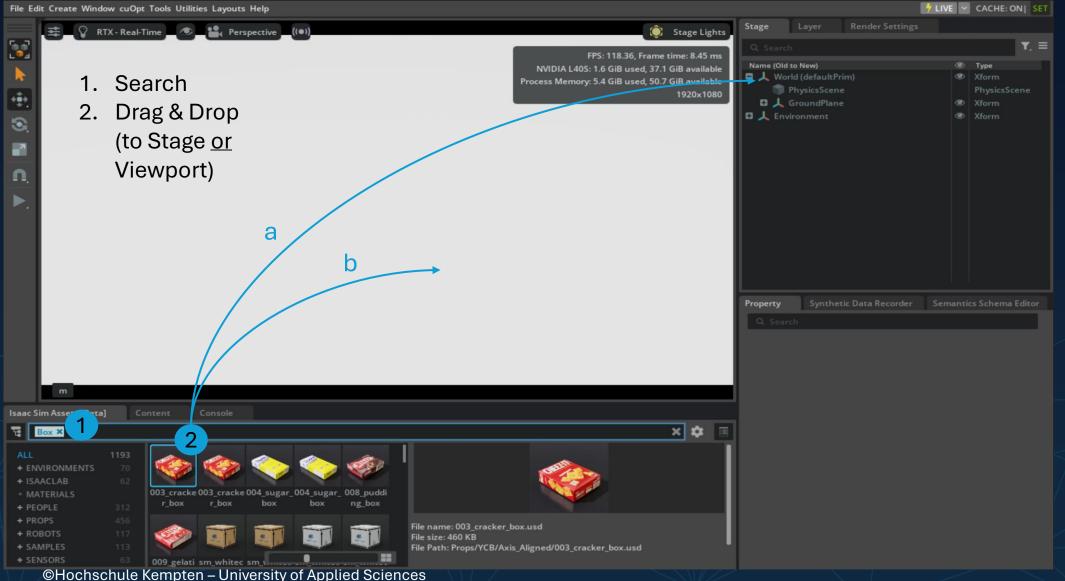
- Import a Box
 - Move above ground plane
 - Adjust the size
 - Add (Rigid Body and and Collider)
 - Press *Play* to start the Simulation



How-to: Use Asset Store



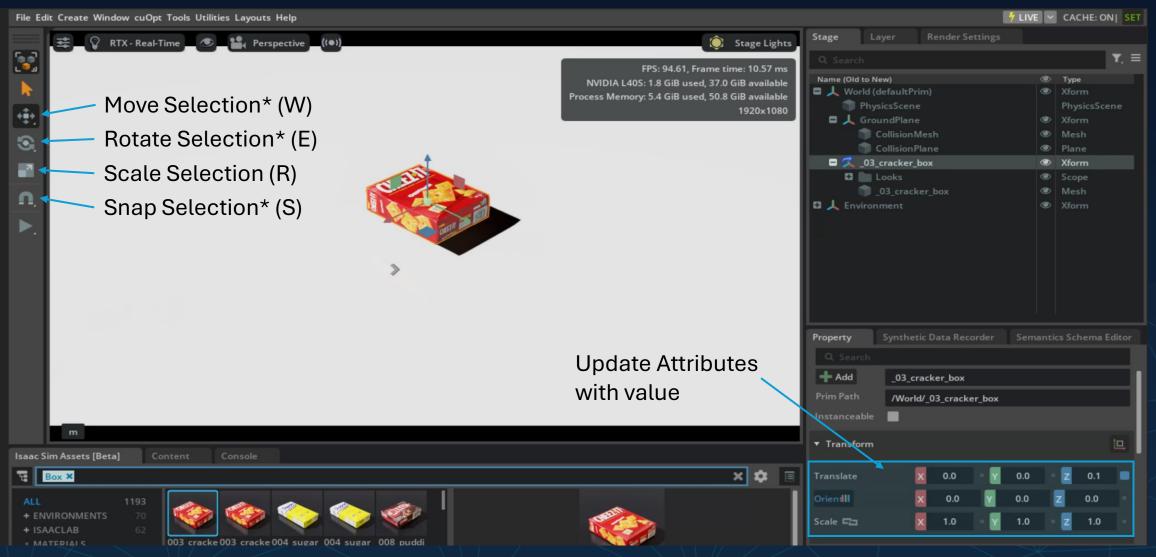
Institut für Produktion und Informatik



How-to: Manipulation in Viewport

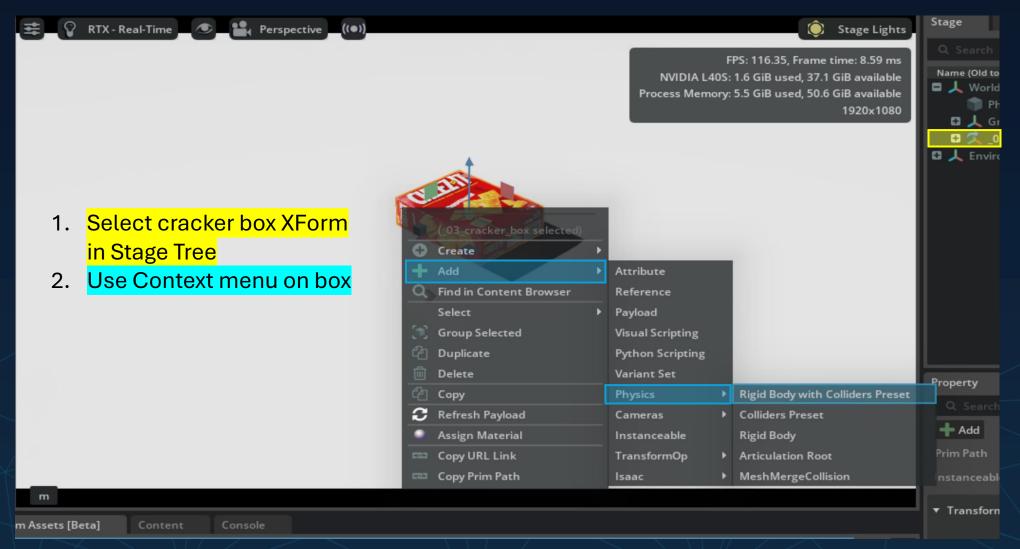
©Hochschule Kempten – University of Applied Sciences





How-to: Add Rigid-body & Collider



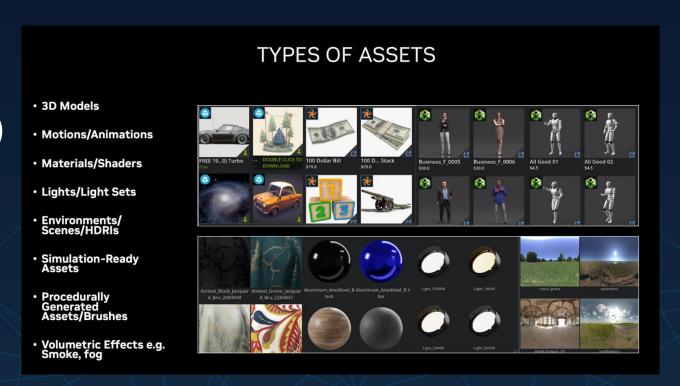








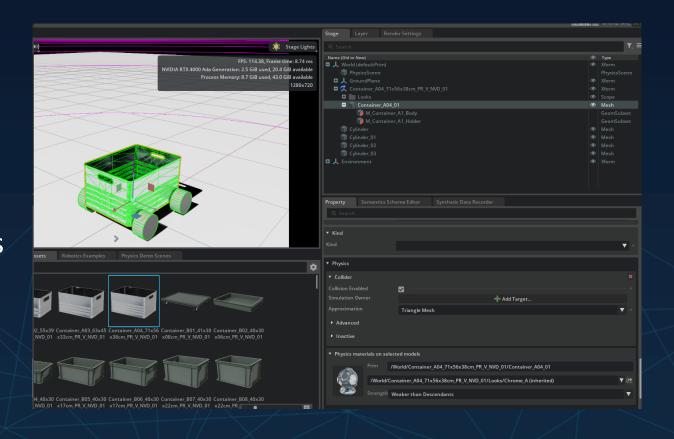
- Import a Box
 - Move above ground plane
 - Adjust the size
 - Add (Rigid Body and and Collider)
 - Press *Play* to start the Simulation





Assemble the first self-built Robot

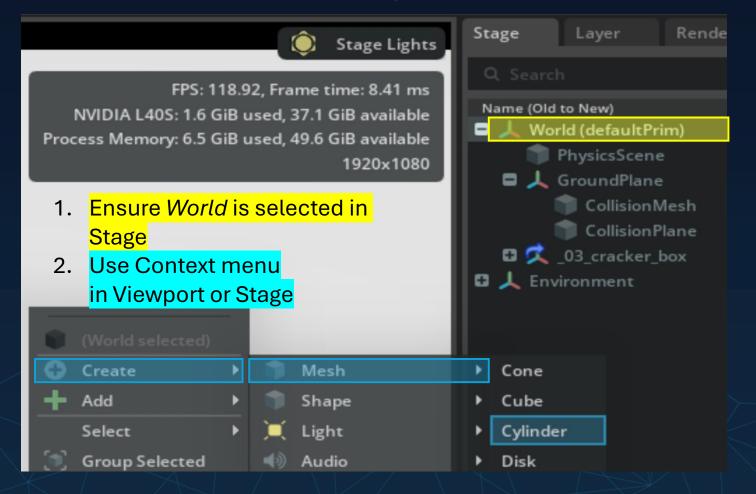
- Stop the Simulation
- Create Cylinders
 - Move/ orient to the side
 - Create a revolute joint
 - Add Angular drive
 - Modify target velocity + stiffness
- Start the Simulation





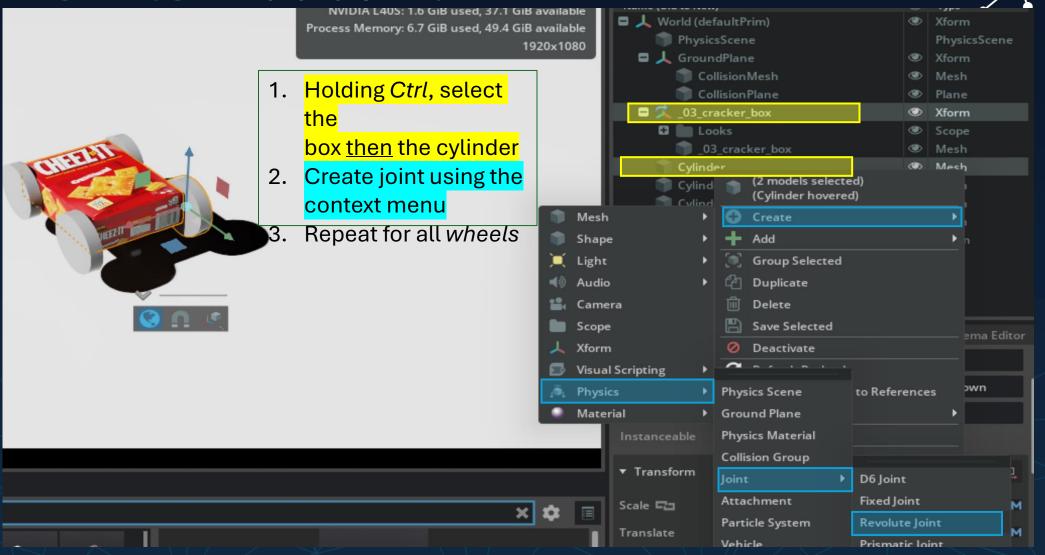
How-to: Add Cylinder





- 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

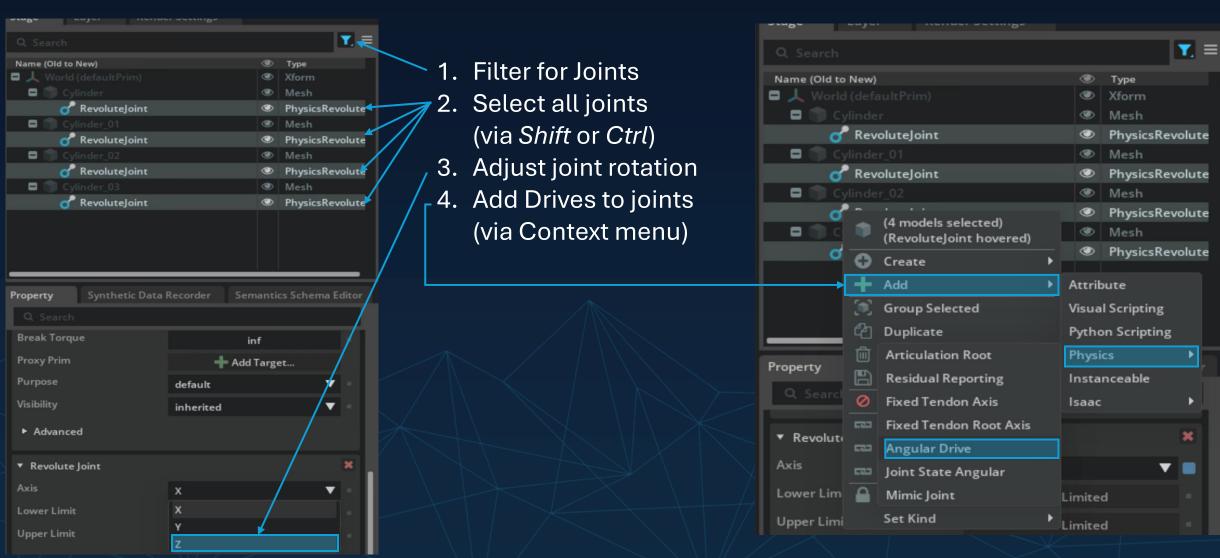
How-to: Add Joint



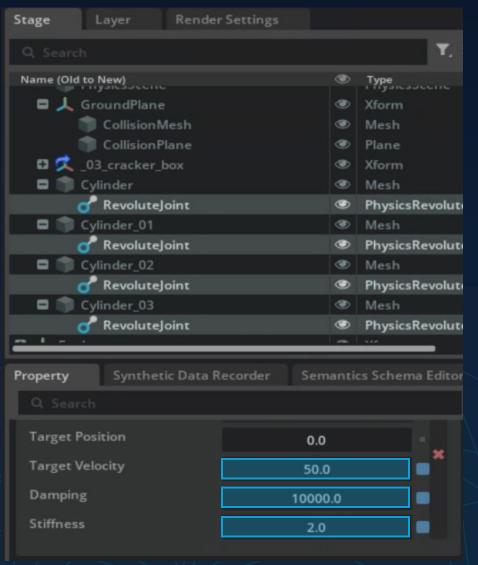


How-to: Adjust Joint and Add Drive





How-to: Adjust Drive



©Hochschule Kempten - University of Applied Sciences



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

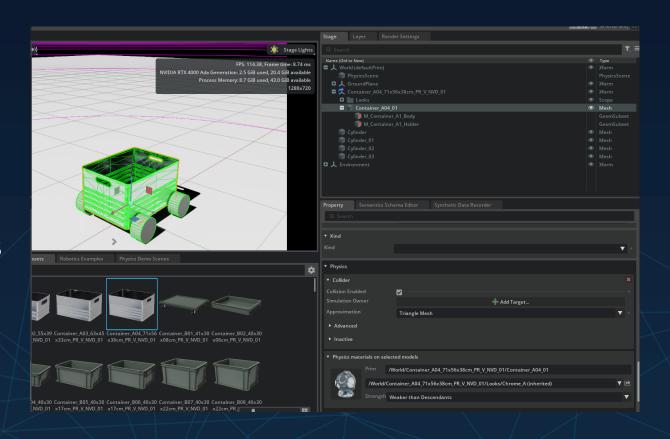
*Not restarting the simulation leads to misleading results due to the forces that already applied before the update





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

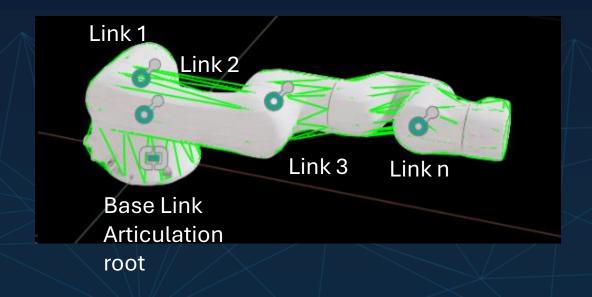




Robot assembly

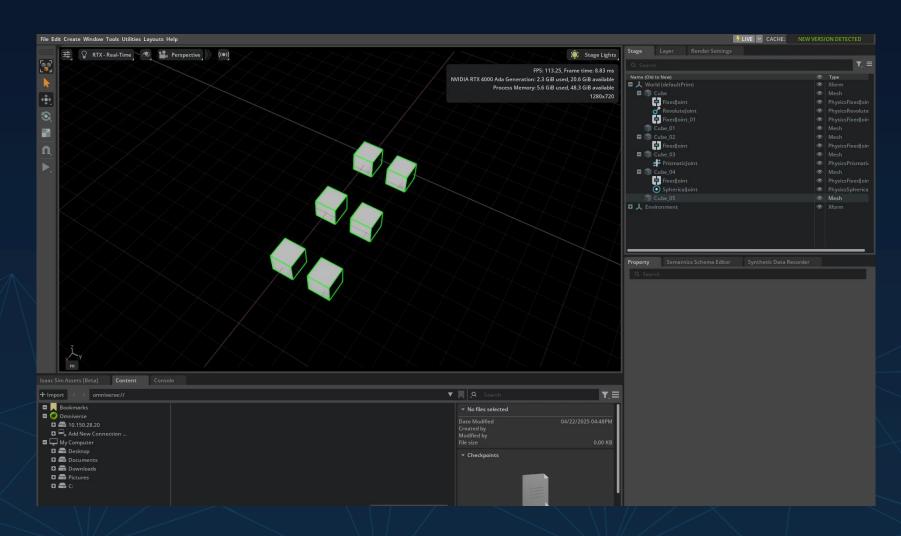


- Structure of mobile / static robot
 - Articulation chain
 - root link free / fixed



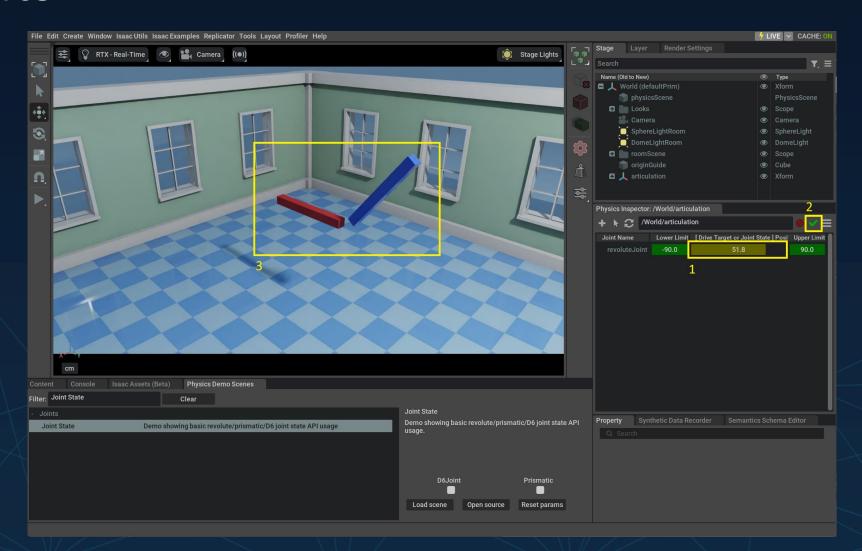


Joint types and properties



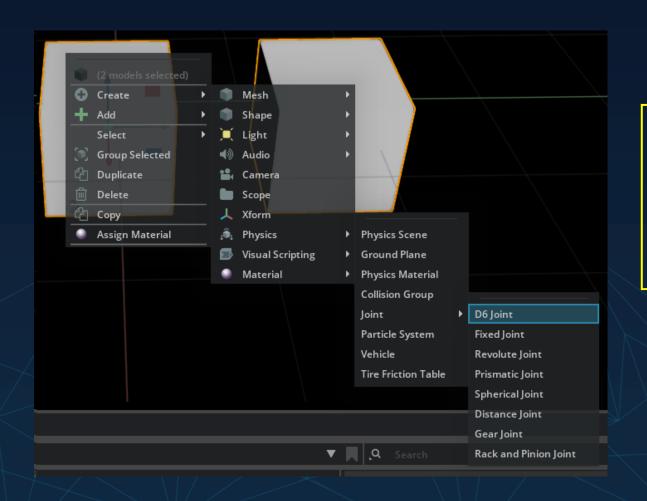


Joints





Joint types and properties



Joint drives are internally simulated with PD-Controllers

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



Until now:

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

10 min Break

To be continued:

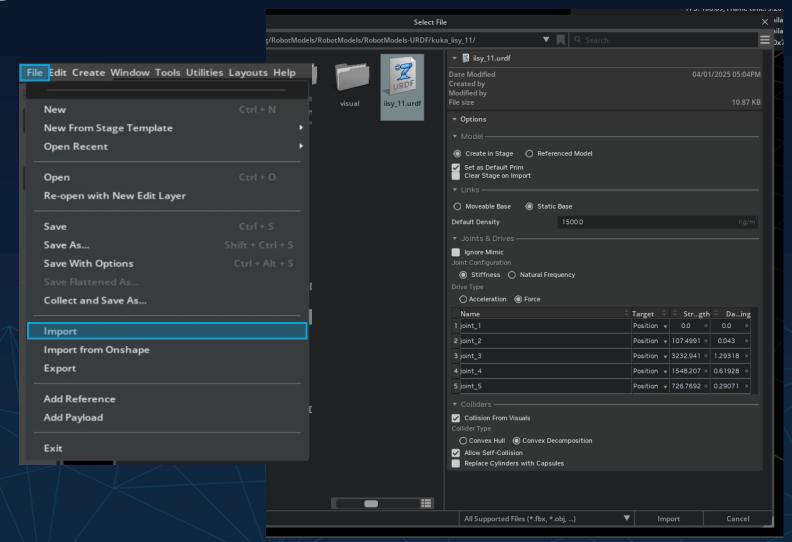
- Practical session:
 - Robot import + tuning



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

How-





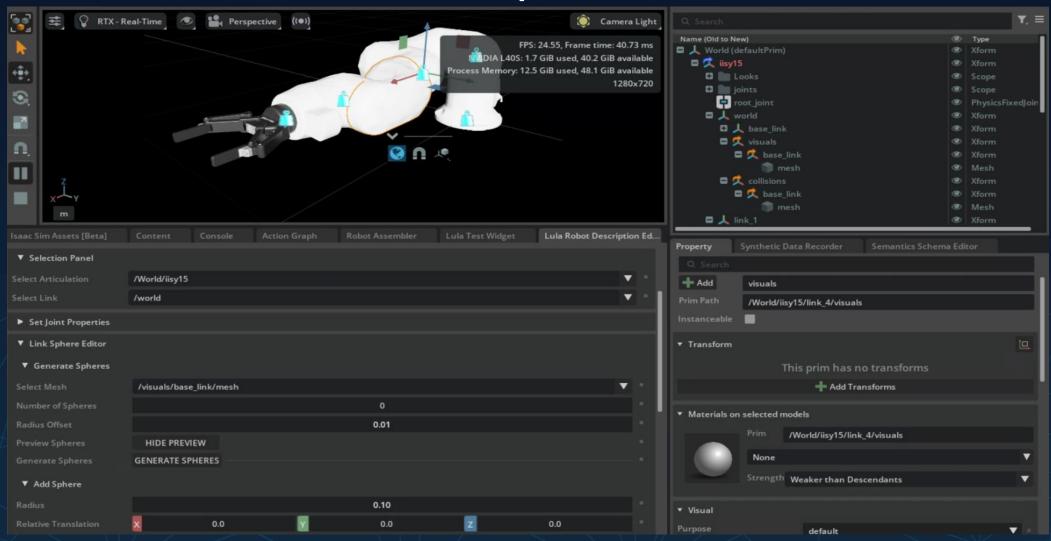
Robot import Tools

- 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



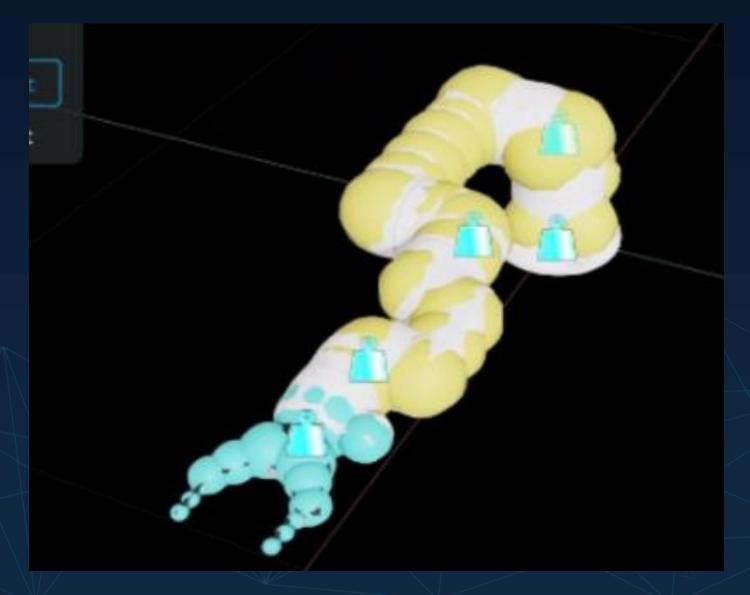


Robot Description Editor



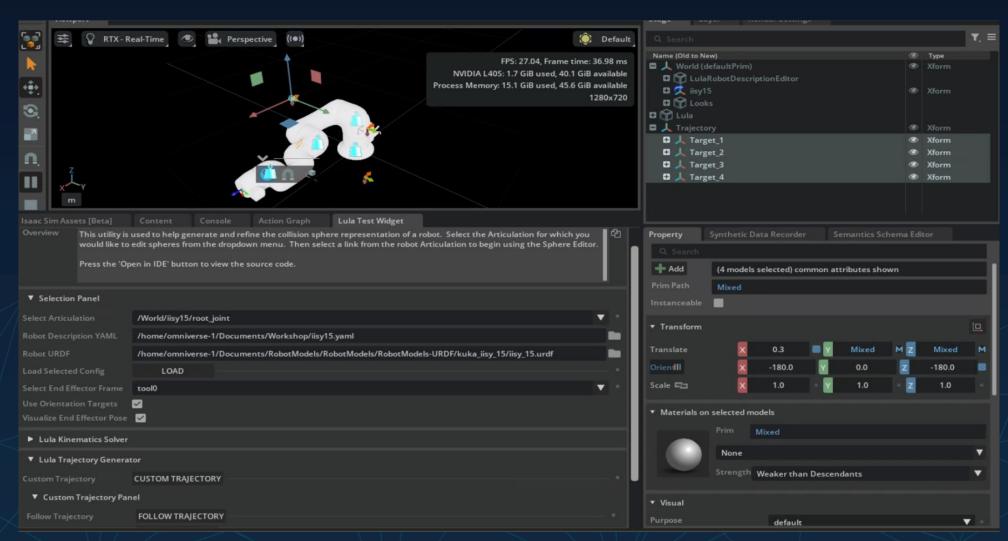


Sphere Approximation Result



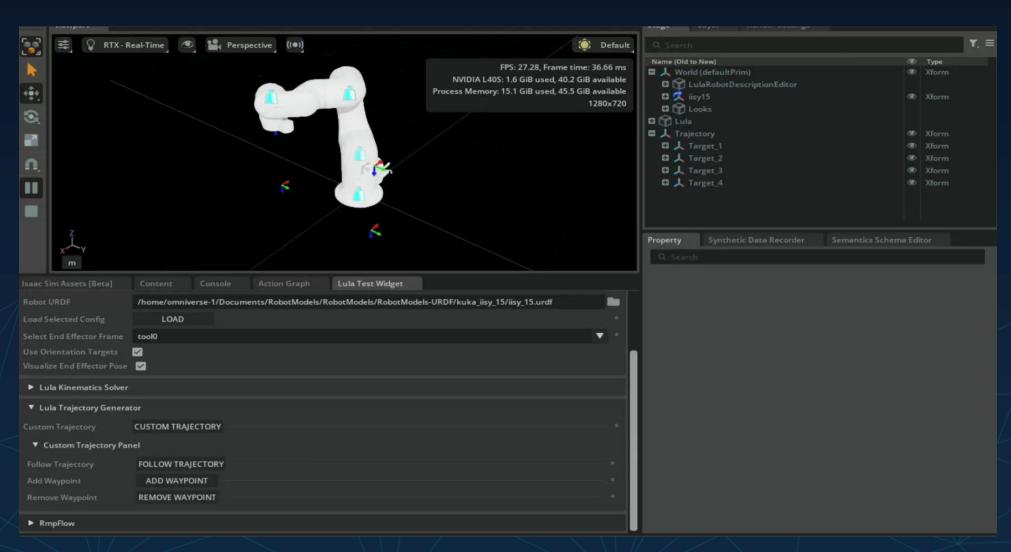














Sensors

- 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