Isaac Sim Surgical Challenge

Last Updated: 8/28/2024

Files Required:

* standard\_surgical\_challenge.usd
* custom\_thread\_2.usd
* phantom\_only\_with\_collisions\_2.usd
* isaac\_sim\_objects (folder)
  + isaac\_sim\_base\_objects.py
  + isaac\_sim\_camera.py
  + isaac\_sim\_psm.py
  + watch\_dog.py
* simulation\_manager.py (changed)
* isaac\_client.py

How to run the Isaac Sim version of the AMBF Surgical Challenge

***Dependencies***

* Isaac Sim

https://docs.omniverse.nvidia.com/isaacsim/latest/installation/index.html

* Nvidia Cache
* AMBF Surgical Challenge

https://github.com/surgical-robotics-ai/surgical\_robotics\_challenge

* ROS1 Noetic

http://wiki.ros.org/noetic/Installation/Ubuntu

* Phantom Omni Drivers or Master Tool Manipulator Drivers

https://github.com/jhu-saw/sawSensablePhantom/blob/main/drivers.md

***Notes:*** The Isaac Sim Surgical Challenge has been developed and tested on Ubuntu 20.04. Isaac Sim itself has several dependencies and hardware requirements to properly run: NVIDIA GPU, 16 GB Ram, and more listed on their website.

***Running Isaac Sim***

The main file to launch the surgical challenge in Isaac Sim can be found in standard\_surgical\_challenge.usd. This file requires the ROS1 extension found in Isaac Sim to work, or be preloaded when launching the application for available ROS controllers. As well, the file requires two supporting files (custom\_thread\_2 and phantom\_only\_with\_collisions\_2) to properly load in the needle, thread, and phantom for the simulation.

*Suggested File Structure*

* custom\_thread\_2.usd
* phantom\_only\_with\_collisions\_2.usd
* <Folder>
  + standard\_surgical\_challenge.usd

|  |
| --- |

If everything loads in properly, you should be greeted by the image above.

***Enabling ROS Control***

To enable ROS1 control of the simulator, you must replace simulation\_manager.py found within AMBF Surgical Robotics Challenge files with the alternative simulation\_manager.py provided with the Isaac Sim Surgical Challenge files. This will allow the user to access Isaac Sim custom object instances to interact with the Isaac Sim ROS stack.

Alongside the simulation\_manger.py, it requires the isaac\_client.py and isaac\_sim\_objects folder and scripts available to run ROS1 with Isaac Sim.

*Suggested File Structure*

From AMBF Surgical Challenge files scripts…

* <scripts>
  + <surgical\_robotics\_challenge>
    - <isaac\_sim\_objects>
    - <pycache>
    - isaac\_sim\_base\_object.py
    - isaac\_sim\_camera.py
    - Isaac\_sim\_psm.py
    - watch\_dog.py
  + …
  + simulation\_manager.py (alternative)
  + isaac\_client.py
  + …

Run Roscore and controller before starting the next step.

Use the following command to then enable ROS1 communication between controller script and Isaac Sim:

**python3 <controller\_script>.py -c isaac\_sim –three false**

***Changing Parameters***

To change physical or visual parameters in the simulation can be done the following ways.

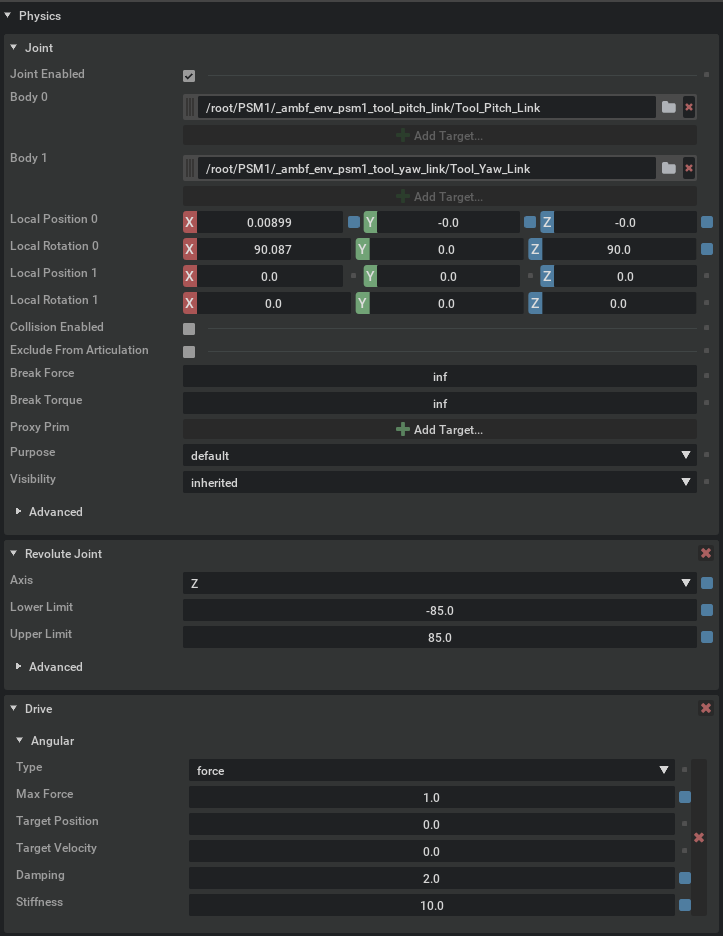
*Changing PSM Parameters*

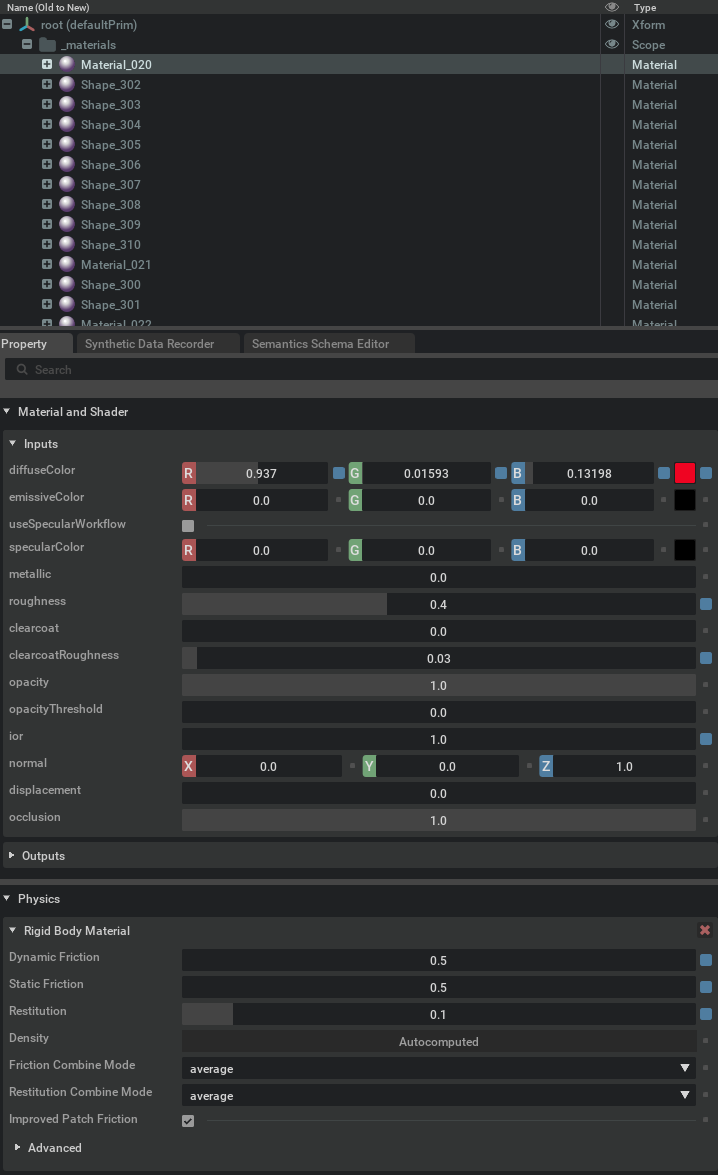
PSM1 and PSM2 objects inside Isaac Sim are defined with rigid bodies and joints. The rigid bodies contain information about rigid body dynamics and mass while joints control movement limits, stiffness, and damping.

Rigid Body Parameters



Joint Parameters



The material properties of certain parts of the PSM or other objects in Isaac Sim can be altered in the following menus depending on which material object is applied to a rigid body.

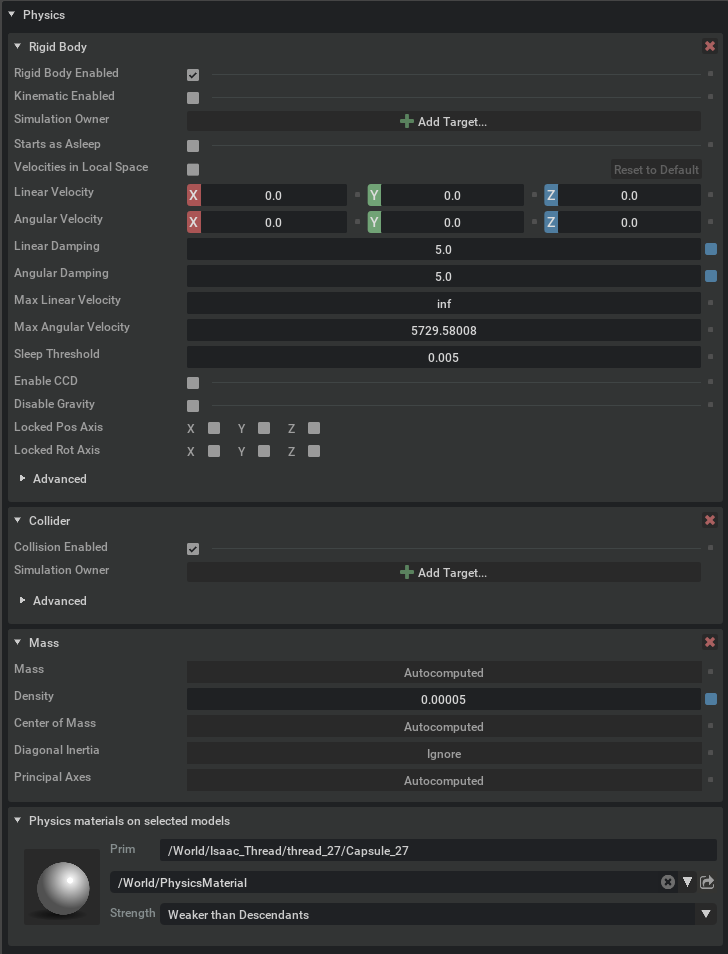
These object materials can be placed on the visual of physic slots available for rigid bodies.



*Changing Thread Parameters*

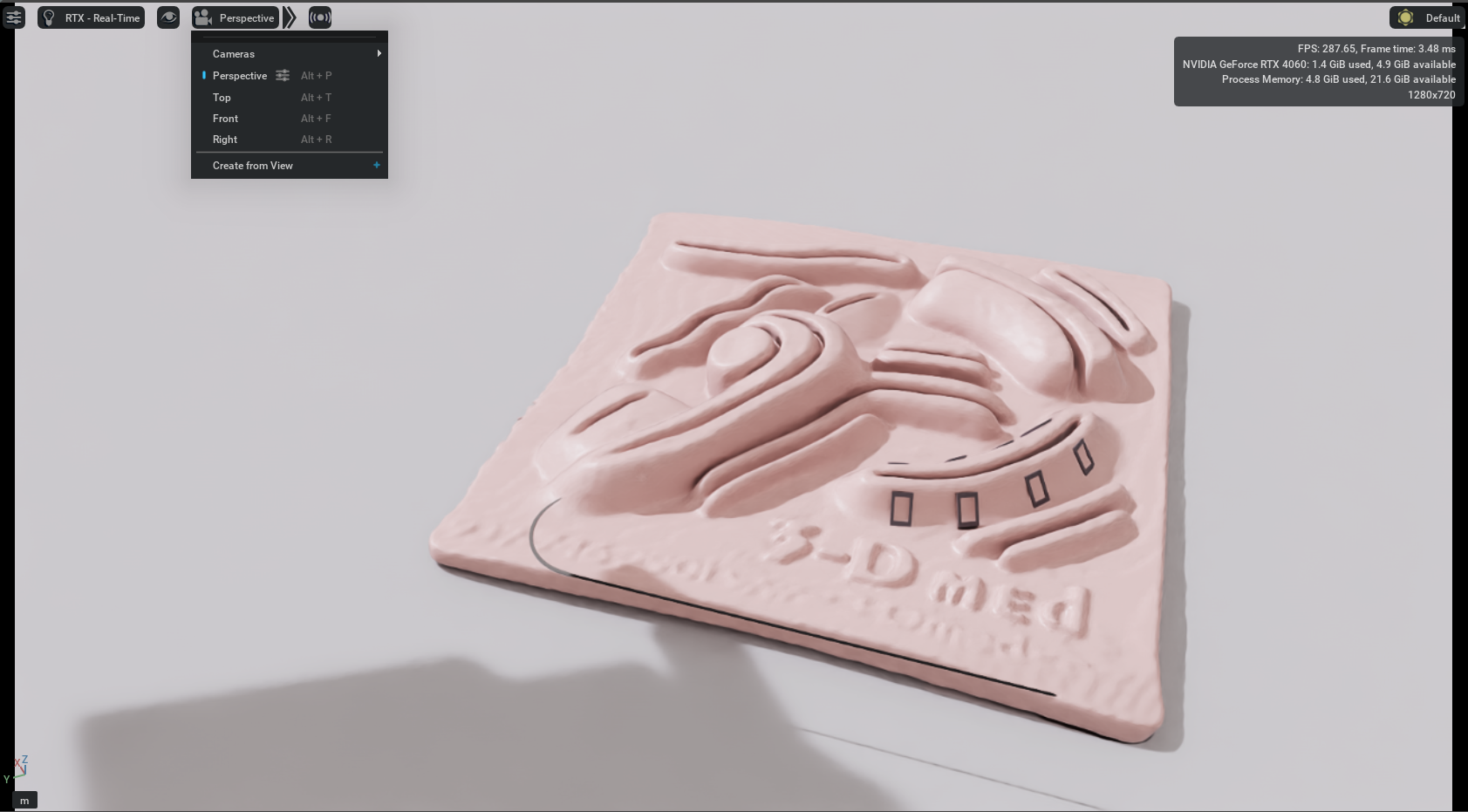
Changing thread parameters requires modifying the referenced objects object in custom\_thread\_2.usd and reloading the object in the standard\_surgical\_challenge.usd. Parameter changes can either involve mass, density, damping, and stiffness.

Thread Physical Properties

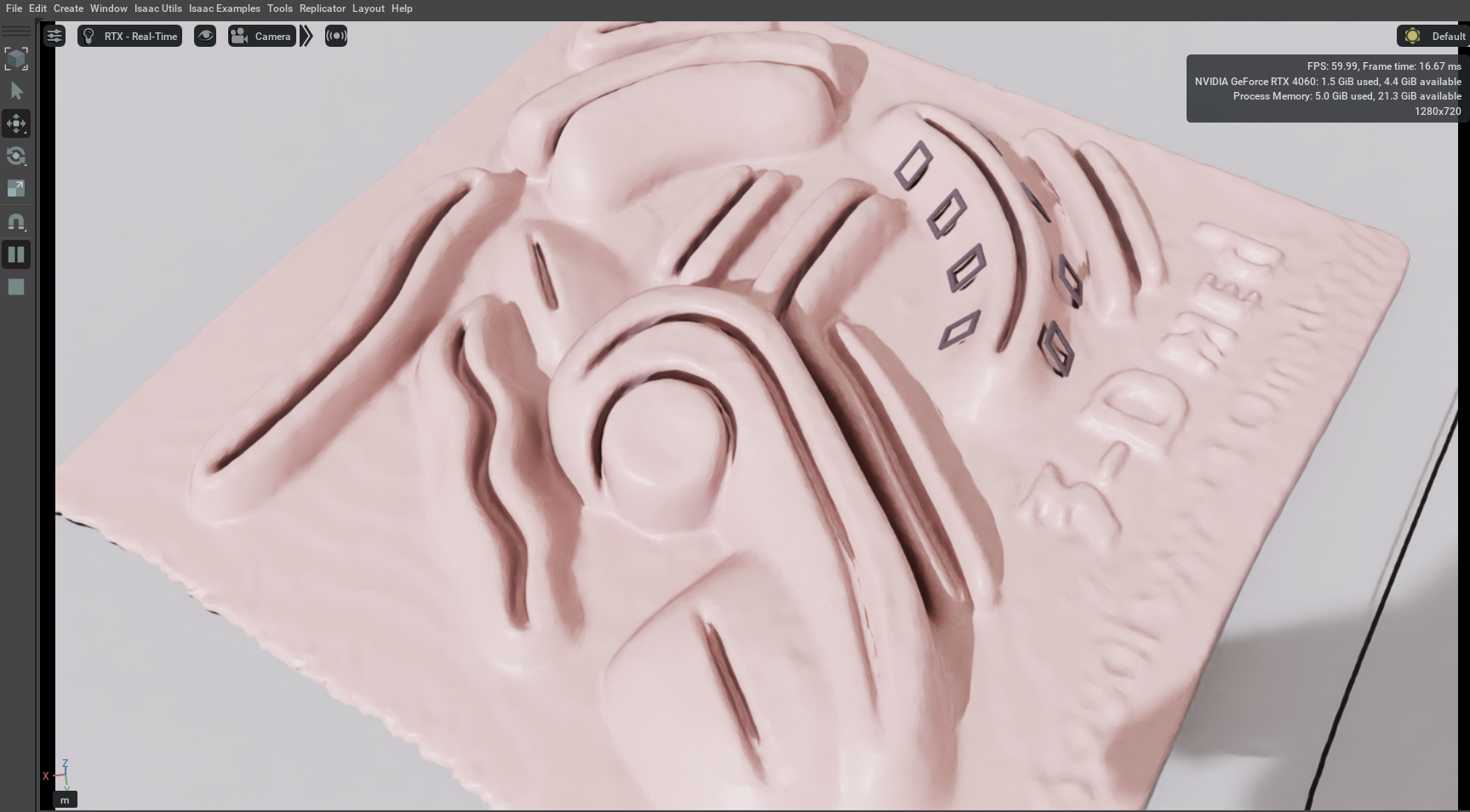


***Running the Surgical Challenge***

1. Open Isaac Sim
2. Open standard\_surgical\_challenge.usd
3. Select camera in view



1. Hit Play Simulation on the left hand side of simulator



1. Run controller python3 -c isaac\_sim –three false