SOFA – Unity3D Asset

V23.12: Online documentation

Contact: contact@infinyTech3D.com

GitHub: https://github.com/InfinyTech3D/SofaUnity

Version 2.0 - 2024-07-26

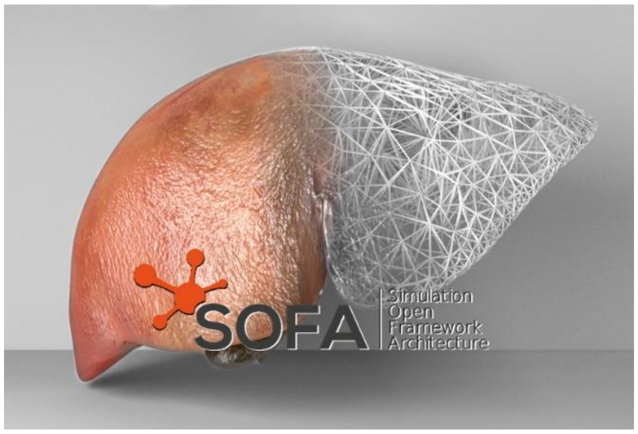
Contents

1. Introduction

1.1 - What is SofaAPAPI-Unity3D ?

The **SOFA Unity3D** integration brings the advanced capabilities of **SOFA** (Simulation Open Framework Architecture) into the **Unity3D** environment, enabling real-time physics simulations. This integration embeds SOFA as a physics engine within Unity3D, allowing developers to harness SOFA's extensive interactivity, flexibility, and performance. With this module, you can create Unity3D applications that include simulations of deformable objects, interactions with haptic devices, and support for VR or AR technologies. Specifically, this integration acts as a bridge between the C++ library **SofaVerseAPI**, which consolidates SOFA concepts into a unified API with multiple levels of integration, and the **SofaUnity** C# asset which exposes this API in **Unity3D**.

1.2 - But what is SOFA ?



SOFA, which stands for Simulation Open Framework Architecture, is a powerful framework designed for the research, prototyping, and development of physics-based simulations. It utilizes advanced scientific approaches such as Finite Element Methods, constraints, numerical methods for ordinary differential equation (ODE) resolution, and precise contact handling for haptic devices. Implemented efficiently in C++, SOFA enables the simulation of both soft and rigid body dynamics, making it possible to create highly realistic and interactive simulations. These capabilities are valuable across various fields, including healthcare, robotics, virtual prototyping, and more, where accurate and dynamic simulation is crucial.

For full description visit: **www.sofa-framework.org**

1.3 - What can I do with it ?

With **SOFA-Unity3D**, you can simulate solid and deformable mechanics, as well as fluid and thermodynamic systems, directly within Unity3D. The integration supports interactions with multiple haptic devices, ensuring continuous constraint resolution and providing a realistic tactile feedback experience. Check the section

Check the section **Features Available** for more details.

1.4 - How to get it ?

The core of the SOFA-Unity3D integration is freely available on the **Unity Asset Store** and can also be accessed via the **InfinyTech3D Github page**. Additional features, such as haptic integration, medical imaging, or pre-configured surgical scenarios, can be acquired on request using the form in the **Request software section**. These specialized assets will soon be available on the Unity Asset Store. For further details or to request additional features, please contact us at: **contact@infinytech3d.com**

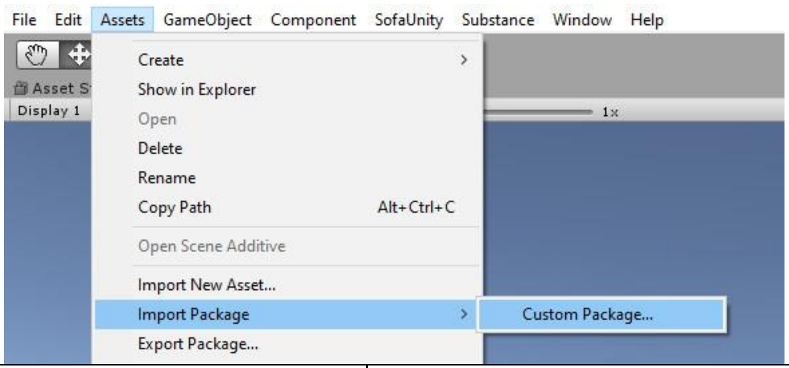
2. Getting started

2.1 – How to install

To install the SOFA-Unity3D asset, follow these steps based on the version you downloaded. Depending, from where First you downloaded it, you might need to Unity3D package and follow those steps:

Downloaded from Unity Asset Store	**Downloaded from Github project page**
1. **Project Setup**: Open your existing Unity project or create a new 3D project. Select the appropriate template (Standard, URP, or HDRP) based on the package version you downloaded.	
2. **Import Package**: In the Unity Editor, navigate to `Assets -> Import Package ->`	

Custom Package and select the downloaded .unitypackage file.



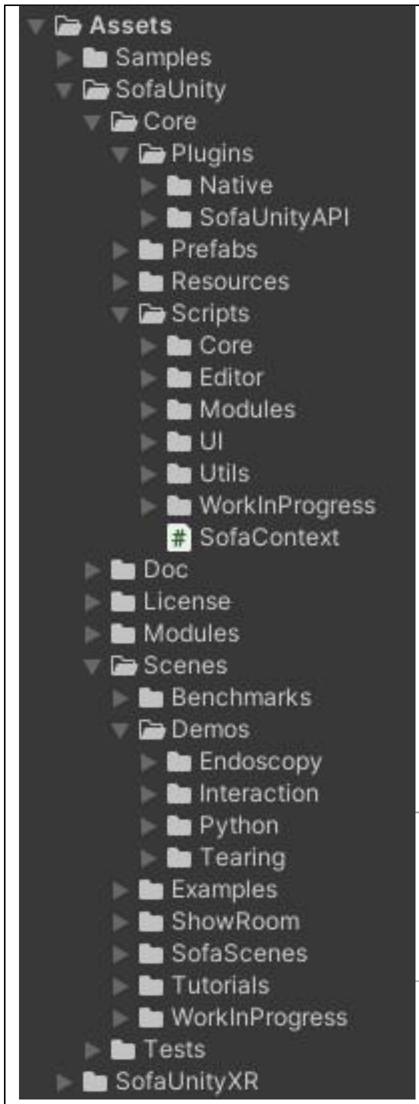
3. **SOFA DLL Files**: Download the SOFA DLL files from the GitHub repository (SofaUnity_v23.12_windows_dll.zip). Extract the contents into the directory {your Unity3D project}/Assets/SofaUnity/Plugins/Native/x64. **Note**: If you downloaded the asset from the GitHub page, this step is unnecessary as the SOFA DLL files are included in the package.	**SOFA DLL Files**: If you downloaded the package: **SofaUnity_v23.12.00_public_withDLL.unitypackage** this step is unnecessary as the SOFA DLL files are included in the package. Otherwise download the file (SofaUnity_v23.12_windows_dll.zip). Extract the contents into the directory {your Unity3D project}/Assets/SofaUnity/Plugins/Native/x64

4. ***For additional Features only: Place the license file you received by email inside {your Unity 3D project}/Assets/SofaUnity/License/****. Replace the existing file if needed.*

5. **Check Config File**: Locate the sofa.ini file in {your Unity3D project}/Assets/SofaUnity/Plugins/Native/x64/.
   - **Update Paths**: Ensure the paths in the config file point to your SOFA scenes and mesh directories.
     The default configuration should look like this:
     ```
     SHARE_DIR={your Unity3D project}/Assets/SofaUnity/scenes/SofaScenes
     EXAMPLES_DIR={your Unity3D project}/Assets/SofaUnity/scenes/SofaScenes
     PYTHON_DIR={your Unity3D project}/Assets/SofaUnity/Core/Plugins/Native/x64/
     ```
   - **Custom Paths**: If you prefer to use different directories for storing SOFA scenes and meshes, update the paths in the sofa.ini file accordingly.

2.2 - Package content

Once the SofaUnity asset is set up, the folder structure should resemble the following layout. Please note that some folders may only appear if you are using the main branch from the GitHub repository and might not be present in the official release versions. Additionally, the `License` and `Modules` folders are included only if you have requested specific assets beyond the core API of SofaUnity.



```
Assets
  Samples
  SofaUnity
    Core
      Plugins
        Native
        SofaUnityAPI
      Prefabs
      Resources
      Scripts
        Core
        Editor
        Modules
        UI
        Utils
        WorkInProgress
        # SofaContext
    Doc
    License
    Modules
    Scenes
      Benchmarks
      Demos
        Endoscopy
        Interaction
        Python
        Tearing
      Examples
      ShowRoom
      SofaScenes
      Tutorials
      WorkInProgress
    Tests
  SofaUnityXR
```

- **Core** → Main folder containing all files of this asset
  - **Plugins**
    - **Native** → SOFA x64 .dll files folder.
    - **SofaUnityAPI** → C# API scripts interfacing with Sofa library.
  - **Prefabs** → Unity prefabs to use SOFA Unity GameObjects
  - **Resources** → Unity materials, textures and images.
  - **Scripts**
    - **Core** → List of classes to represent SOFA components hiearchy using Unity3D GameObject.
    - **Editor** → Classes to implement SOFA components Unity Editor inspectors.
    - **Modules** → Specific classes inheriting from the Core classes
    - **UI** → 3D UI scripts to control SOFA player or components in Game.
    - **Utils** → List of scripts of useful components/algorithms to use SOFA simulation.

- **Doc** → Offline documentation (prefer online version which is more up-to-date).

- **License** → Folder to store the license. Only for specific assets (Modules).

- **Modules** → Folder containing the different advanced features as additional assets.

- **Scenes** → Folder containing the different advanced features as additional assets.
  - **Benchmarks** → Some specific Unity scene to benchmark SOFA components or integration
  - **Demos** → Advanced simulation scenes to demonstrate SOFA capabilities. Such as endoscopy, carving, cutting, python scripting and tearing simulation.
  - **Examples** → All basic examples to demonstrate SOFA integration in Unity.

	◦ **SofaScenes** → All SOFA scene files in .scn or .py with their data. ◦ **Tutorials** → Work in progress tutorials, prefer using the examples.
	• **Tests** → Folder containing non-regression scene test API.

2.3 – Check Installation

After setting up the SofaUnity asset, the simplest way to verify a successful installation is by checking the Unity console for errors when you open one of the basic example scenes located in `Scenes/Examples/`, such as `Example_01_SimpleLiver.unity` or `Example_02_Caduceus.unity`. If no errors are present, proceed by playing the scene. You should observe the 3D model moving by default, and the movement should vary when you adjust the gravity value in the SofaContext UI inspector.

3. Main principles

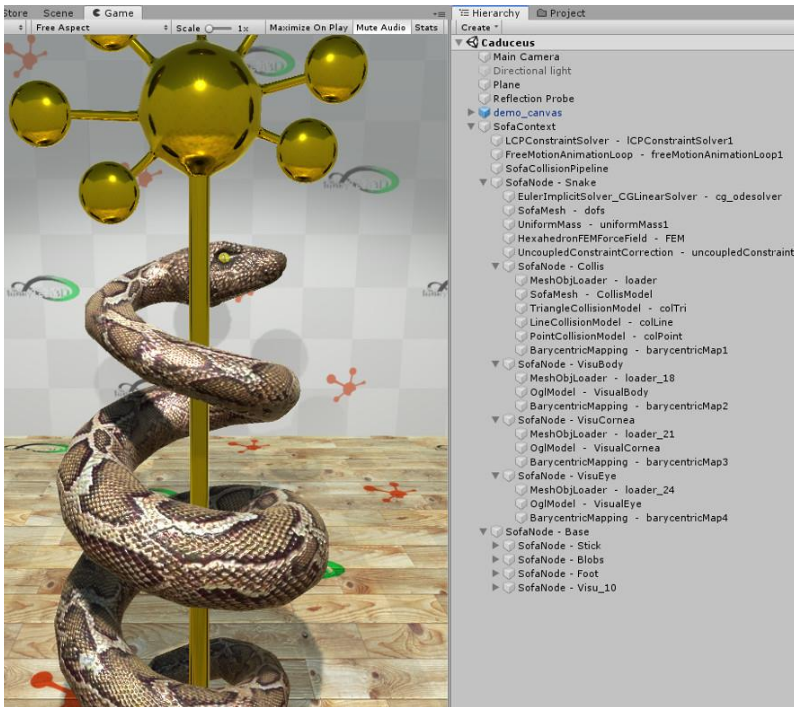
To effectively use the SOFA-Unity3D integration, it's essential to have a basic understanding of both the Unity3D Editor and the SOFA physics engine. For more comprehensive documentation and tutorials, refer to the **Unity Manual** and the **SOFA documentation**. For further assistance, including training opportunities, consider contacting the SOFA consortium.

3.1 – Technical Details

The SOFA-Unity3D integration enables running SOFA simulations in the background while your Unity project is active. It offers a full integration with two-way communication between Unity and SOFA. Here are the key features and technical aspects:

- **Simulation Execution:** During each Unity update cycle, a SOFA simulation step is computed. However, this execution occurs asynchronously, meaning that the simulation's time step may not match Unity's update frequency.

- **Hierarchy and Mapping:** The integration uses a hierarchy of C# classes based on Unity's MonoBehaviour system. This maps SOFA components to Unity GameObjects, allowing for scene recreation and interaction within Unity. The SOFA simulation scene is parsed and represented using these GameObjects, which can be edited through the Unity Inspector view. Parameters and configurations are saved as part of the scene graph accessible via the Hierarchy view.



```
▼ Caduceus
    Main Camera
    Directional light
    Plane
    Reflection Probe
  ▶ demo_canvas
  ▼ SofaContext
      LCPConstraintSolver - lCPConstraintSolver1
      FreeMotionAnimationLoop - freeMotionAnimationLoop1
      SofaCollisionPipeline
    ▼ SofaNode - Snake
        EulerImplicitSolver_CGLinearSolver - cg_odesolver
        SofaMesh - dofs
        UniformMass - uniformMass1
        HexahedronFEMForceField - FEM
        UncoupledConstraintCorrection - uncoupledConstraint
      ▼ SofaNode - Collis
          MeshObjLoader - loader
          SofaMesh - CollisModel
          TriangleCollisionModel - colTri
          LineCollisionModel - colLine
          PointCollisionModel - colPoint
          BarycentricMapping - barycentricMap1
      ▼ SofaNode - VisuBody
          MeshObjLoader - loader_18
          OglModel - VisualBody
          BarycentricMapping - barycentricMap2
      ▼ SofaNode - VisuCornea
          MeshObjLoader - loader_21
          OglModel - VisualCornea
          BarycentricMapping - barycentricMap3
      ▼ SofaNode - VisuEye
          MeshObjLoader - loader_24
          OglModel - VisualEye
          BarycentricMapping - barycentricMap4
    ▼ SofaNode - Base
      ▶ SofaNode - Stick
      ▶ SofaNode - Blobs
      ▶ SofaNode - Foot
      ▶ SofaNode - Visu_10
```

- **Scenes:** Additionally, the package includes several basic examples demonstrating SOFA's capabilities and advanced demos for interacting with deformable objects, controlling objects on the SOFA side, and deploying a catheter.

3.2 – Limitations

SOFA integrates a wide range of physics principles, leading to potentially complex simulation scene graphs.

1. **Knowledge Requirement:** SOFA integrates a wide range of physics principles, leading to potentially complex simulation scene graphs. Users need to be familiar with creating SOFA simulation scenes, as the asset does not currently support high-level SOFA-Unity object creation or direct SOFA component creation within the Unity3D editor. These interactions are still a work in progress and can be approached in several ways:

    - **Loading Existing SOFA Scenes**: The simplest integration method involves loading pre-existing SOFA simulation scenes (*.scn files). For more information, refer to the **SOFA Scene Parsing** section.
    - **Creating High-Level SOFA-Unity Objects**: Users can create high-level objects that bridge SOFA and Unity. Details on this process are available in the **Object Creation** section.
    - **Direct Creation of SOFA Components in Unity**: Advanced users can directly represent SOFA components within the Unity3D scene graph. Guidance on this approach can be found in the **Graph Edition** section.

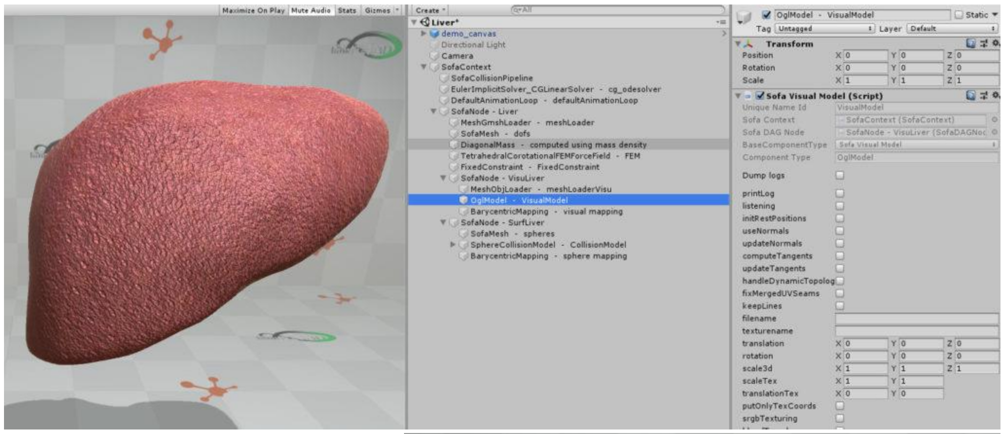
2. **Specialized Applications**: This asset serves as a foundational toolkit and does not include pre-configured solutions for specialized applications such as medical training, robotics research, or virtual prototyping.
3. **Platform Support**: The Windows DLLs required for SOFA must be downloaded separately, as they are not included in the asset. Additionally, there is currently no support for Linux, Mac, or Android platforms.

SofaComponent specialization

As explain earlier in the **SOFA Scene Parsing section.** Several categories of component from **SOFA** are mapped as **Unity3D** specialized GameObject. The others are created as **SofaComponent** generic GameObject.

Those specialized implementation allows for example to see the multi-model representation of a 3D object. In the Liver scene it is possible to see the visual

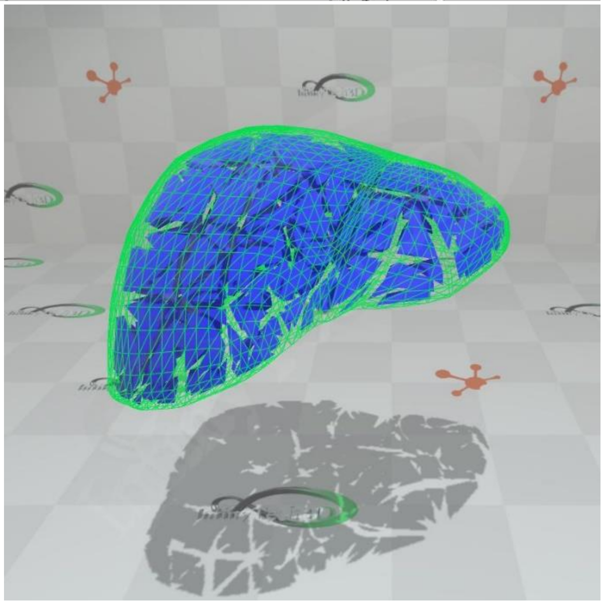
Classic visual rendering of the object using MeshFilter inside the component SofaVisualModel.



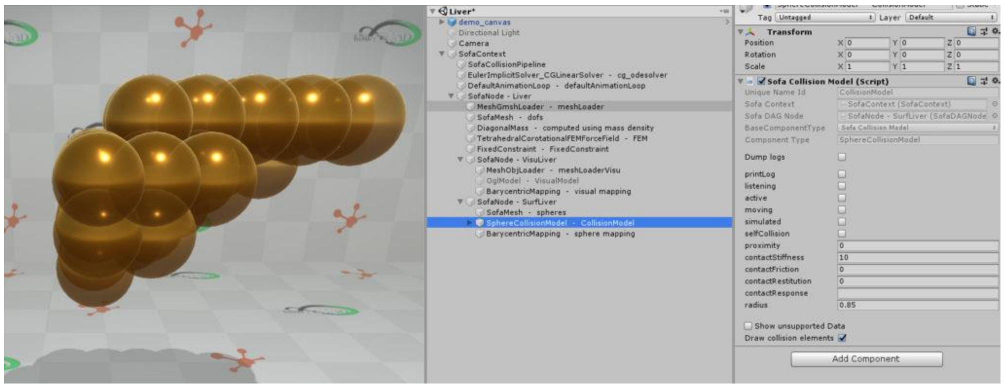
```
▼ Liver*
  ▶ demo_canvas
    Directional Light
    Camera
  ▼ SofaContext
      SofaCollisionPipeline
      EulerImplicitSolver_CGLinearSolver  -  cg_odesolver
      DefaultAnimationLoop  -  defaultAnimationLoop
    ▼ SofaNode - Liver
        MeshGmshLoader  -  meshLoader
        SofaMesh  -  dofs
        DiagonalMass  -  computed using mass density
        TetrahedralCorotationalFEMForceField  -  FEM
        FixedConstraint  -  FixedConstraint
      ▼ SofaNode - VisuLiver
          MeshObjLoader  -  meshLoaderVisu
          OglModel  -  VisualModel
          BarycentricMapping  -  visual mapping
      ▼ SofaNode - SurfLiver
          SofaMesh  -  spheres
        ▶ SphereCollisionModel  -  CollisionModel
          BarycentricMapping  -  sphere mapping
```

```
☑ OglModel  -  VisualModel                 ☐ Static
Tag  Untagged                    Layer  Default
▼ Transform
Position               X 0      Y 0      Z 0
Rotation               X 0      Y 0      Z 0
Scale                  X 1      Y 1      Z 1
▼ ☑ Sofa Visual Model (Script)
Unique Name Id         VisualModel
Sofa Context           SofaContext (SofaContext)
Sofa DAG Node          SofaNode - VisuLiver (SofaDAGNoc
BaseComponentType      Sofa Visual Model
Component Type         OglModel
Dump logs              ☐
printLog               ☐
listening              ☐
initRestPositions      ☐
useNormals             ☐
updateNormals          ☐
computeTangents        ☐
updateTangents         ☐
handleDynamicTopolog   ☐
fixMergedUVSeams       ☐
keepLines              ☐
filename
texturename
translation            X 0      Y 0      Z 0
rotation               X 0      Y 0      Z 0
scale3d                X 1      Y 1      Z 1
scaleTex               X 1      Y 1
translationTex         X 0      Y 0
putOnlyTexCoords       ☐
srgbTexturing          ☐
```

FEM element can be displayed inside the SofaFEMForceField by activating the Mesh Renderer. Note that it can slow down the simulation as for example the tetrahedron in this case need to be updated at each frame.



Collision model can also be displayed in the SofaCollisionModel by activating the option **Draw collision elements**.
Note that same as for FEM, this can slow down the simulation as the element need to be updated at each frame.

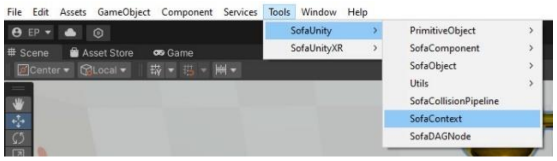


```
▼ Liver*
  ▶ demo_canvas
    Directional Light
    Camera
  ▼ SofaContext
      SofaCollisionPipeline
      EulerImplicitSolver_CGLinearSolver  -  cg_odesolver
      DefaultAnimationLoop  -  defaultAnimationLoop
    ▼ SofaNode - Liver
        MeshGmshLoader  -  meshLoader
        SofaMesh  -  dofs
        DiagonalMass  -  computed using mass density
        TetrahedralCorotationalFEMForceField  -  FEM
        FixedConstraint  -  FixedConstraint
      ▼ SofaNode - VisuLiver
          MeshObjLoader  -  meshLoaderVisu
          OglModel  -  VisualModel
          BarycentricMapping  -  visual mapping
      ▼ SofaNode - SurfLiver
          SofaMesh  -  spheres
        ▶ SphereCollisionModel  -  CollisionModel
          BarycentricMapping  -  sphere mapping
```

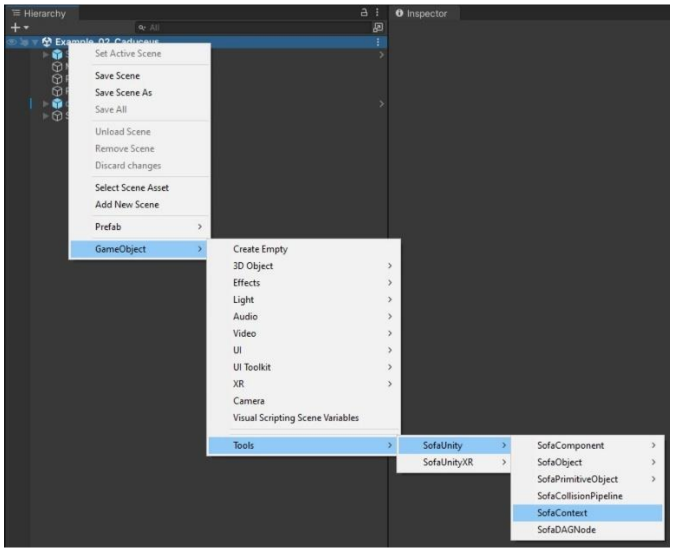
```
Tag  Untagged                    Layer  Default
▼ Transform
Position               X 0      Y 0      Z 0
Rotation               X 0      Y 0      Z 0
Scale                  X 1      Y 1      Z 1
▼ ☑ Sofa Collision Model (Script)
Unique Name Id         CollisionModel
Sofa Context           SofaContext (SofaContext)
Sofa DAG Node          SofaNode - SurfLiver (SofaDAGNode
BaseComponentType      Sofa Collision Model
Component Type         SphereCollisionModel
Dump logs              ☐
printLog               ☐
listening              ☐
active                 ☐
moving                 ☐
simulated              ☐
selfCollision          ☐
proximity              0
contactStiffness       10
contactFriction        0
contactRestitution     0
contactResponse
radius                 0.85
☐ Show unsupported Data
Draw collision elements ☑
```

3.3 – SofaContext Creation

To utilize the **SOFA** physics engine within **Unity3D**, the first essential step is to create a `SofaContext` object. This `Unity3D GameObject` acts as the root of the SOFA simulation, managing the simulation's operations in the background. All other objects that are part of the SOFA simulation will be children of this root object. You can create the `SofaContext` object by using the top menu within SofaUnity or by right-clicking in the Hierarchy panel.



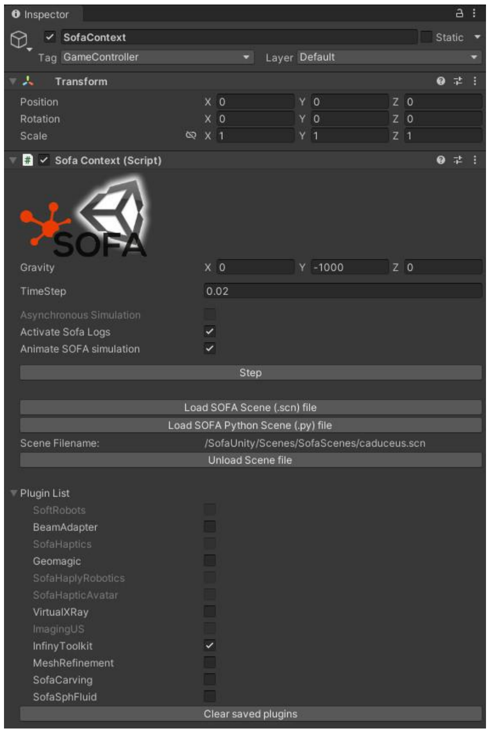
You can create the `SofaContext` object by using the top menu within `Tools/SofaUnity` or by right-clicking in the Hierarchy panel.



<u>SofaContext inspector</u>

The Inspector of the `SofaContext` GameObject offers several key functionalities:

- **Gravity and Timestep Adjustment**: Allows you to modify the gravity settings and the simulation timestep directly from the Inspector.
- **Message Handling**: Provides options to enable a message handler, which lets you receive SOFA logs, including information, warnings, and errors.
- **Manual Simulation Control**: Enables you to disable automatic simulation and use the step button to manually progress through the simulation.
- **Loading SOFA Scenes**: Supports loading SOFA scene files in both `.scn` and Python formats.
- **SOFA Plugin Management**: Includes a list of SOFA plugins that can be loaded. **Important Note:** Plugins must be selected before loading any scene that requires them.



```
SofaContext
Tag GameController          Layer Default
Position                 X 0     Y 0      Z 0
Rotation                 X 0     Y 0      Z 0
Scale                    X 1     Y 1      Z 1
Gravity                  X 0     Y -1000  Z 0
TimeStep                 0.02
Asynchronous Simulation  ☐
Activate Sofa Logs       ✓
Animate SOFA simulation  ✓
Scene Filename:          /SofaUnity/Scenes/SofaScenes/caduceus.scn
Plugin List
  SoftRobots             ☐
  BeamAdapter            ☐
  SofaHaptics            ☐
  Geomagic               ☐
  SofaHaplyRobotics      ☐
  SofaHapticAvatar       ☐
  VirtualXRay            ☐
  ImagingUS              ☐
  InfinyToolkit          ✓
  MeshRefinement         ☐
  SofaCarving            ☐
  SofaSphFluid           ☐
```

When a `SofaContext` is created, it automatically generates a root `SOFADAGNode` and a default `AnimationLoop` GameObject, which are crucial for setting up and controlling the simulation environment.

Lastly, the Transform component of the Unity `GameObject` containing the `SofaContext` can be used to position and scale the **SOFA** frame (3D world) within the Unity 3D scene.

SOFA components to SofaUnity GameObjects

Since the version 1.0 of SofaAPAPI-Unity3D several components from **SOFA** are mapped as **Unity3D** specialized GameObject. The others are created as **SofaComponent** generic GameObject. The generic implementation of **SofaComponent** only allows to interact with the **SOFA** component parameters. Those parameters correspond to the **SOFA Data** of a component. Whereas the component specialization allow to add more complexe behavior. Like for example the display of FEM or collision model. For more details, check section **Scene Graph edition**

Here is the list of **Unity3D** specific GameObject used to define **SOFA** simulation scene.component categories particularly handled in :

SofaAnimationLoop

This GameObject correspond to **SOFA AnimationLoop**. This is a key component in charge of ruling the simulation and the mechanical resolution system.
See the corresponding documentation on **SOFA** webiste: **AnimationLoop**

SofaCollisionModel

This GameObject correspond to type of collision element used in **SOFA** to compute primitive intersection. It is usually either Spheres, Triangle, Edges or Points. This component is directly linked to the SofaMesh.
See the corresponding documentation on **SOFA** webiste: **IntersectionMethod**

SofaConstraint

This GameObject will describe some constraint applyed to the current system. It is only there to help describing the simulation scene. It could be either projective or lagrange constraints.
check more information regarding constraint on **SOFA** webiste: **Projective Constraint**

SofaFEMForceField

his GameObject correspond to **SOFA AnimationLoop**. This is a key component in charge of ruling the simulation and the mechanical resolution system.
See the corresponding documentation on **SOFA** webiste: **AnimationLoop**

SofaLoader

This GameObject is for the moment only used to display the type of mesh used. Several formats are handled in **SOFA**. The more known are: obj, vtk, stl or gmsh.

SofaMass

This GameObject correspond to the Mass component applied to this mesh. This component can be used to change the total mass of the object or its mass density.
More information regarding Mass can be found on **SOFA** webiste: **Mass**

SofaMechanicalMapping

This GameObject is used to inform if a mechanical mapping is used between different mesh object. The link to the input and ouput mesh will be displayed.
This is a particular mechanism of **SOFA** which is explained into details here: **mappings**

SofaMesh

This GameObject is one of the main component. It corresponds to the mesh used in used **SOFA**. This component will retrieve the number of points as well as their positions in 3D space but also the complete topology of the object.
This GameOject is the main container used to fill MeshFilter or other algorithms. More information regarding Topology can be found on **SOFA** webiste: **topology**

SofaSolver

This GameObject is used to inform which type of integration scheme is used as well as the linear solver used to resolve it.
More information regarding the mechanical system resolution can be fond on **SOFA** website in the **Integration Schemes section** and **Linear Solvers section**

SofaVisualModel

This GameObject is used to map **SOFA** output position from an OglModel/VisualModel to **Unity3D** rendering. A MeshFilter and a Mesh renderer will be created and linked to **SOFA** side.

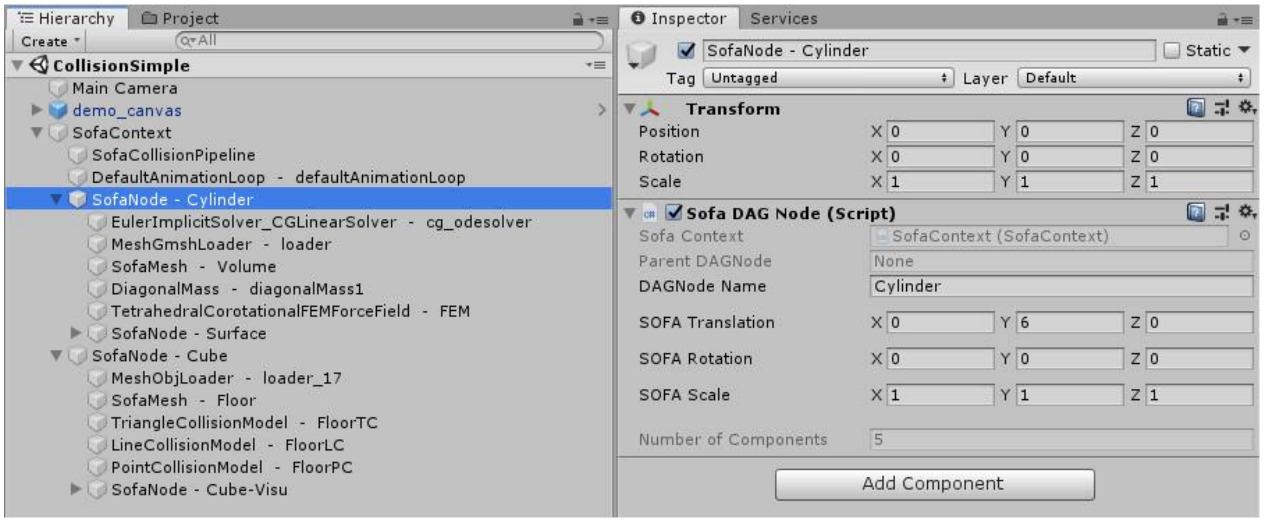
SofaDAGNode scope

SOFA simulation is using a *Directed Acyclic Graph* concept to design hiearchy structures, similar to what can be seen in **Unity3D** Hierarchy panel. The GameObjects SofaDAGNode are used to retrieve this scene Node architecture.

It is highly advise to be familliar with the **SOFA** scene graph structure **descrived here.**

For example in the scene **CollisionSimple**. The root Node is embedded inside the GameObject SofaContext and the two others objects are identified by the SofaNode: Cube and Cylinder.

Finally, the SofaDAGNode inspector allows to see the number of components handled and allows also to translate, rotate and scale the object in the SOFA world.



```
CollisionSimple
  Main Camera
  ▶ demo_canvas
  ▼ SofaContext
      SofaCollisionPipeline
      DefaultAnimationLoop - defaultAnimationLoop
    ▼ SofaNode - Cylinder
        EulerImplicitSolver_CGLinearSolver - cg_odesolver
        MeshGmshLoader - loader
        SofaMesh - Volume
        DiagonalMass - diagonalMass1
        TetrahedralCorotationalFEMForceField - FEM
      ▶ SofaNode - Surface
    ▼ SofaNode - Cube
        MeshObjLoader - loader_17
        SofaMesh - Floor
        TriangleCollisionModel - FloorTC
        LineCollisionModel - FloorLC
        PointCollisionModel - FloorPC
      ▶ SofaNode - Cube-Visu
```

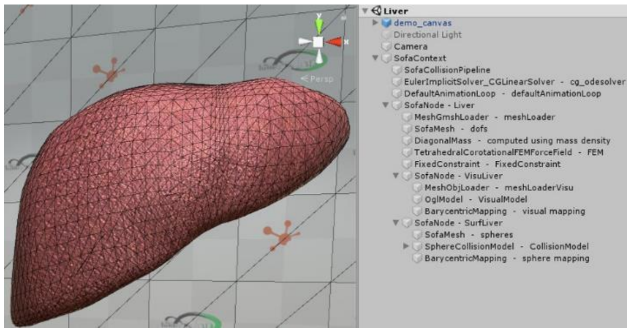
```
☑ SofaNode - Cylinder                         ☐ Static
Tag  Untagged          Layer  Default

Transform
Position      X 0    Y 0    Z 0
Rotation      X 0    Y 0    Z 0
Scale         X 1    Y 1    Z 1

☑ Sofa DAG Node (Script)
Sofa Context        SofaContext (SofaContext)
Parent DAGNode      None
DAGNode Name        Cylinder

SOFA Translation    X 0    Y 6    Z 0
SOFA Rotation       X 0    Y 0    Z 0
SOFA Scale          X 1    Y 1    Z 1

Number of Components   5
```

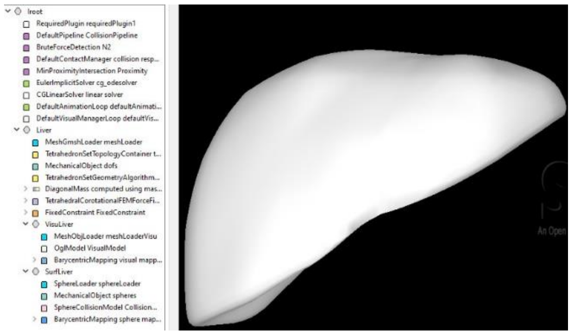
SOFA scene parsing

First approach to create a **SOFA** simulation scene inside **Unity3D** is to load an existing SOFA simulation scene. This is an XML file with .scn extension describing the graph architecture and the components.

First thing to do is always to create the **SofaContext GameObject**. Then, use the **Load Scene button** to import a scn file. Note that the gravity and timestep parameters will be overwritten by the values loaded from the scene file. But it is possible change those value again after the load. Last thing important to notice is that Unity and Sofa have the X axis inverted. Thus to have the same view of the scene from Sofa and Unity, Scale "-1 1 1" should be applyed in the SofaContext.

Once loaded, every Node of the **SOFA** graph will be transposed as SofaDAGNode and the component into SofaUnity GameObjects.


Liver
demo_canvas
Directional Light
Camera
SofaContext
SofaCollisionPipeline
EulerImplicitSolver_CGLinearSolver - cg_odesolver
DefaultAnimationLoop - defaultAnimationLoop
SofaNode - Liver
MeshGmshLoader - meshLoader
SofaMesh - dofs
DiagonalMass - computed using mass density
TetrahedralCorotationalFEMForceField - FEM
FixedConstraint - FixedConstraint
SofaNode - VisuLiver
MeshObjLoader - meshLoaderVisu
OglModel - VisualModel
BarycentricMapping - visual mapping
SofaNode - SurfLiver
SofaMesh - spheres
SphereCollisionModel - CollisionModel
BarycentricMapping - sphere mapping
Liver scene loaded inside Unity3D with the hieararchy display


Same Liver scene loaded inside runSofa GUI

3.2 - SOFA Object creation

Like in the **SOFA** simulation scene parsing. The first step to do is to create the **SofaContext GameObject**.
From that step, it is now possible to create several *SOFA Objects* through the UI Hierarchy panel.
It can be seen as a higher API level which will create all the **SOFA** DAGNode and components to create the object.
The first type of objects are simple primitive like cubes, cylinders or planes. The second type are specific SOFA objects like collision objects.

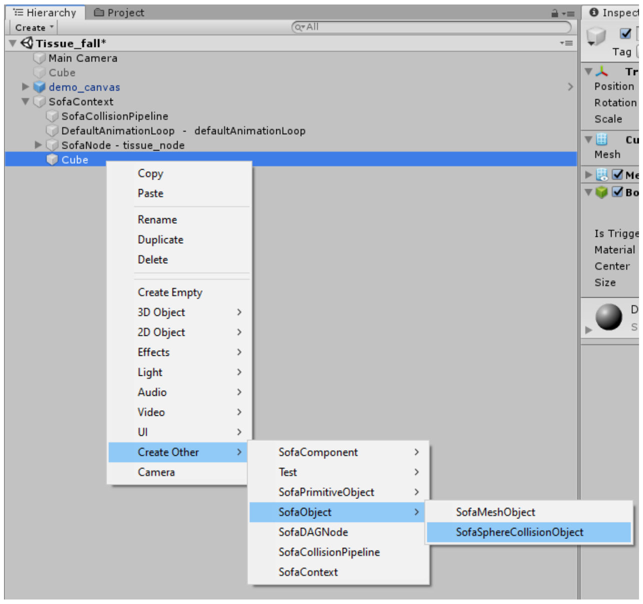
This feature is still under development and might not work as expected.

In addition to the SofaPrimitiveObject, some specific objects will be provided. For the moment this section is very limited.

The first noticeable object is the possibility to create a sphere collision model on top of **Unity3D** objects.

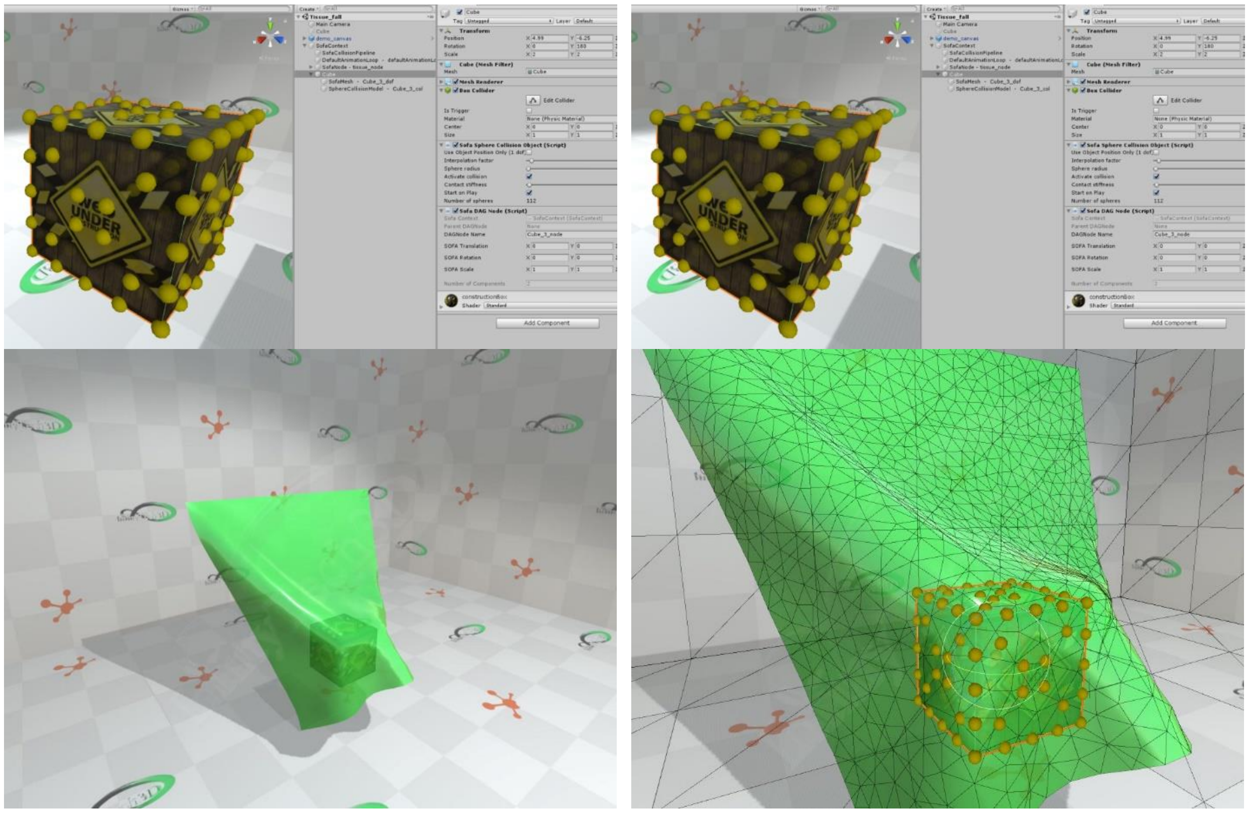
The **SofaSphereCollisionModel** will create spheres that can be seen as gizmos on the Unity side which are mapped to collision sphere inside the Sofa simulation scene.

The demo scene *Tissue_fall* is an example of that feature.



Several parameters can be set:

- **Interpolation factor:** a factor to compute the number of sphere to create on the given geometry.
- **Sphere radius:** radius of the collision spheres.
- **Activate collision:** collide or not with the other objects.
- **Contact stiffness:** the force of the repulsion used by the collision sphere.



3.3 – SOFA Graph Edition

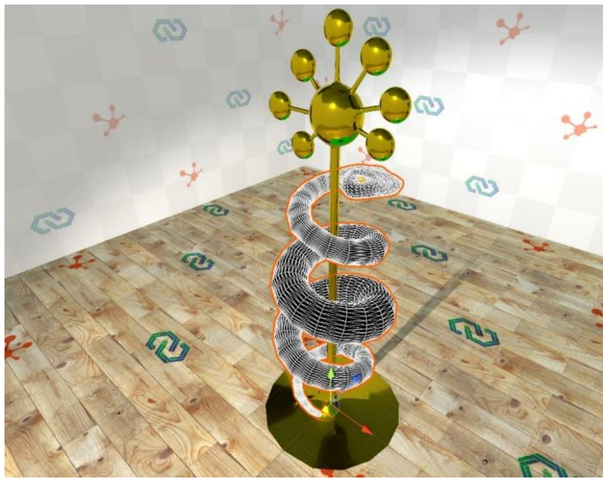
This API is still work in progress. It correspond to the deeper level of API where all edition between SofaUnity components will be possible. It can be seen as an alternative to the scene file or the python scripts usually used by SOFA users.

For the moment those features should only be used to finalise the simulation graph of a loaded scene. Several components, described below, are already available and it is also possible to manually add new SofaDAGNode in the simulation graph.

4. Available Features

The **SofaUnity** Assets can be used to perform simple physical simulation of deformable 3D models in Unity to more advanced mechanical behavior. Below is a summary of all examples and demo scene available in this package. Full information can be found online at the different links.

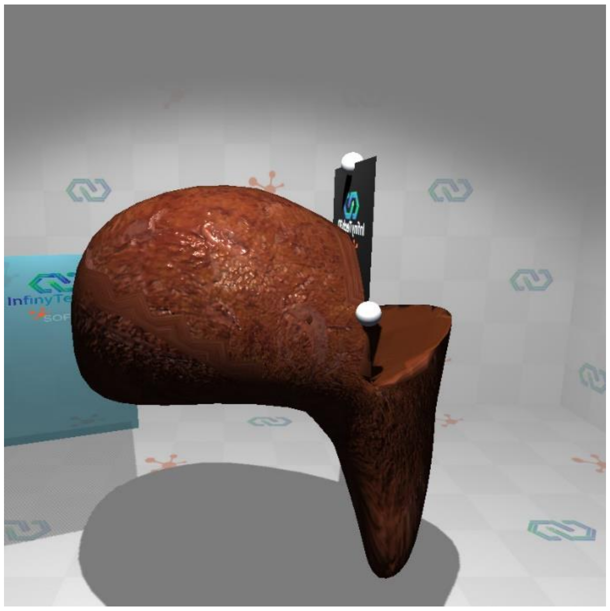
Physics – Basic Examples



This section describes the scenes located in ***SofaUnity/Scenes/Examples/***. These scenes illustrate basic physics that can be simulated using the SOFA framework in Unity. Most of them are examples present as is in the SOFA repository.

Full documentation online

Physics – Simulation interactions

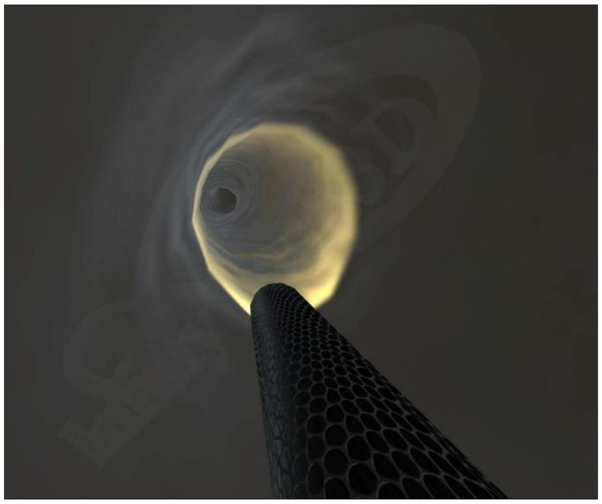


This section describes the scenes located in ***SofaUnity/Scenes/Demos/Interaction***. These scenes illustrate more advanced physics simulation scene with Unity interaction either using the mouse, the keyboard or user interface. The demos described below are:

1. Demo_01_LiverInteraction
2. Demo_02_LiverInteraction-HD
3. Demo_03_CubeCarving
4. Demo_04_CubeAdvancedCarving
5. Demo_05_LiverCut
6. Demo_06_LiverCut-CUDA

Full documentation online

Endoscopy - SOFA Beam Adapter plugin



This section illustrate the scenes located in ***/Scenes/Demos/Endoscopy/BeamAdapter/***

Those scene integrate the SOFA BeamAdapter plugin allowing to have a dynamic implementation of the FEM beam elements. More details can be found on github.

In those examples, Unity is used to control the BeamAdapter navigation and also attach camera at the tip of the tool.

Full documentation online

Endoscopy – Virtual capsule navigation

To use the capsule navigation system, a single rigid degree of freedom is simulated on the SOFA side. This is what is done in the ***"SofaNode – Capsule"*** Node. The SofaMesh is composed of a single position without mesh and a sphere collision.

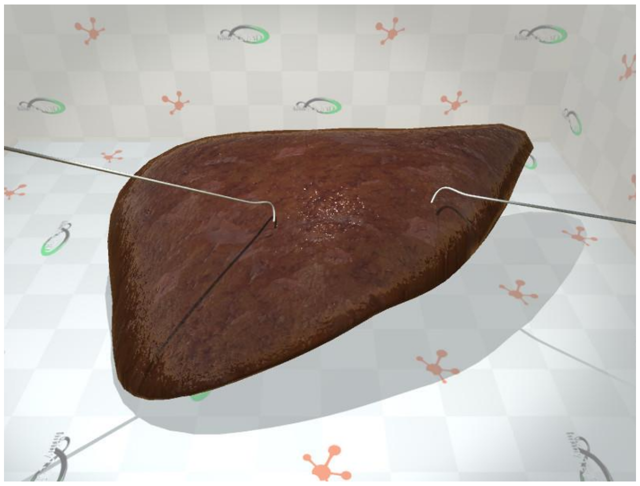
On the Unity side, a 3D object is used such as a capsule, but any 3D mesh could be used. The ***SofaCapsuleController*** script is used on that object to link the position of the SOFA rigid object with this Unity GameObject Transform.

Full documentation online

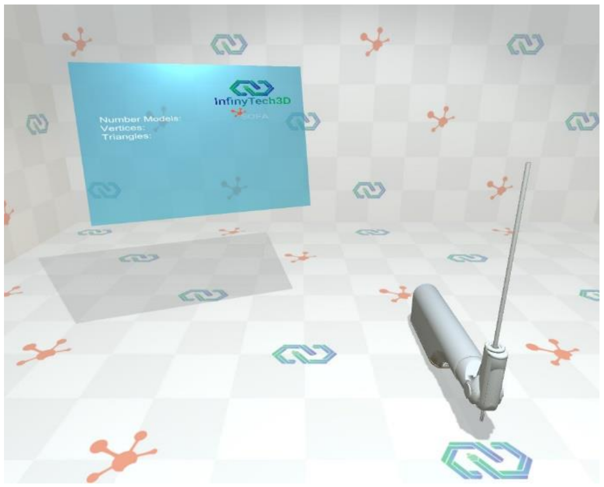
5. Additional Assets

This section gives a quick overview of additional assets adding specific features to the SOFA Unity integration. Those assets can be purchased on the Unity Assets Store or requested from this online form.

Haptic device simulations



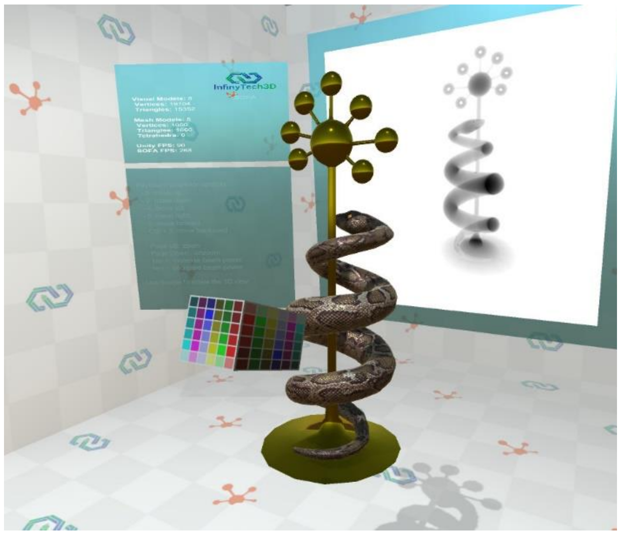
This asset allows to use SOFA Geomagic plugin inside Unity3D



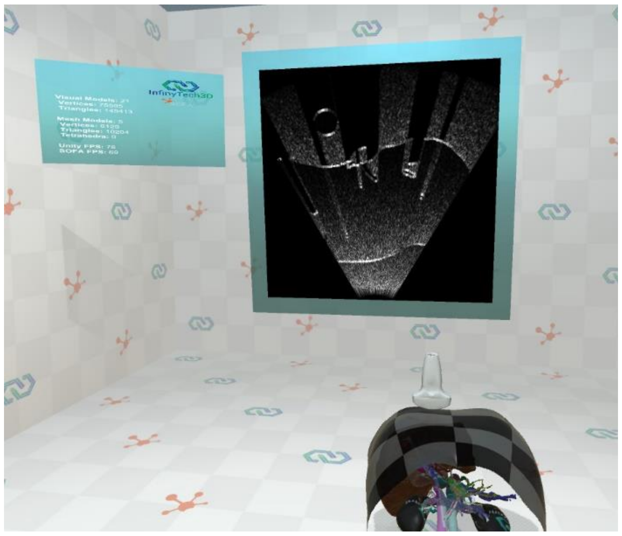
This asset integrates the SOFA Follou Haptic Avatar plugin

This asset integrates the SOFA Haply Robotics Inverse3 plugin

Medical Imaging rendering



This asset allows to compute the fluoroscopic images of 3D models inside SOFA in real time while the simulation is running



This asset allows to compute UltraSound images of 3D models inside SOFA in real time while the simulation is running