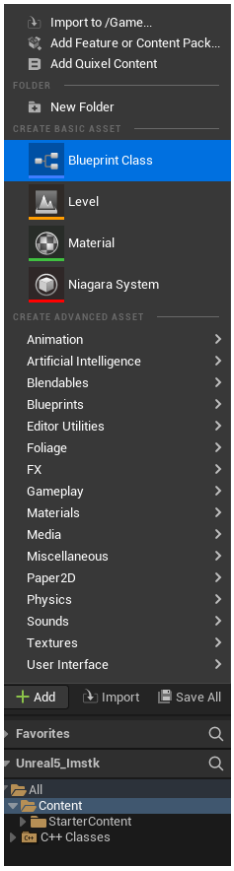
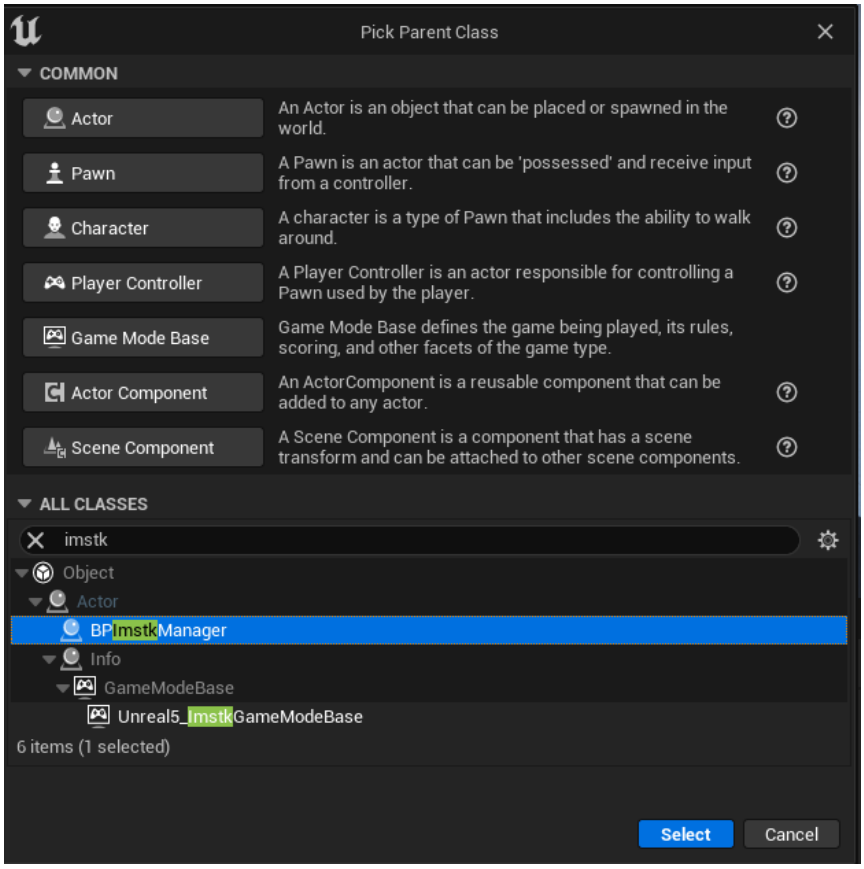
iMSTK Unreal Engine Plugin

Getting Started

1. Create a folder called Plugins if it doesn't already exist in the directory of your .uproject file
2. Extract Imstk.zip into the Plugins folder and launch unreal
3. For each map where iMSTK will be used, create a blueprint that is a child of "BPImstkManager"





4. Drag the blueprint into the map

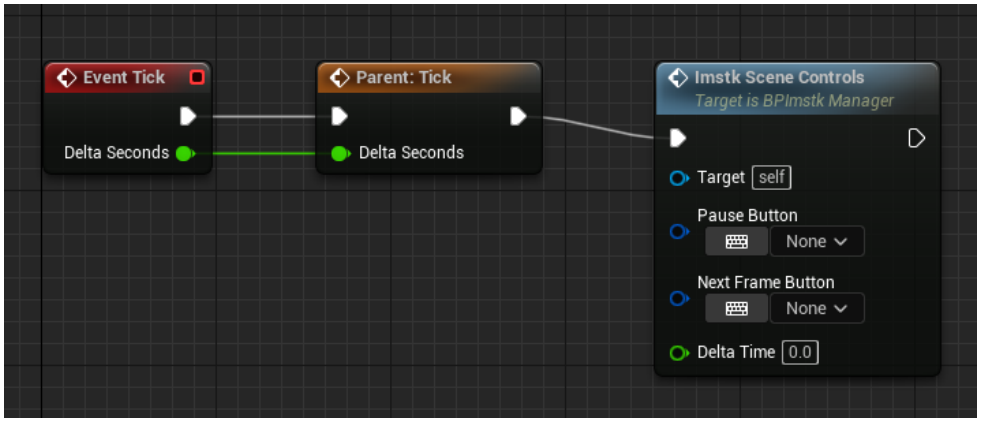
5. Open the blueprint and add the node FinalizeImstk after Parent:BeginPlay

iMSTK Scene Settings

1. Gravity, time step and other values for the iMSTK scene can be set from the manager blueprint
2. Scene Scale will reduce the size of the scene in iMSTK by that degree (default is 100 since one Unreal unit is 1cm and one iMSTK unit is 1m)

Adding Controls to iMSTK

1. Open the IMSTK manager blueprint
2. Add the node “ImstkSceneControls” after Parent:Tick



3. Assign the buttons and delta time values

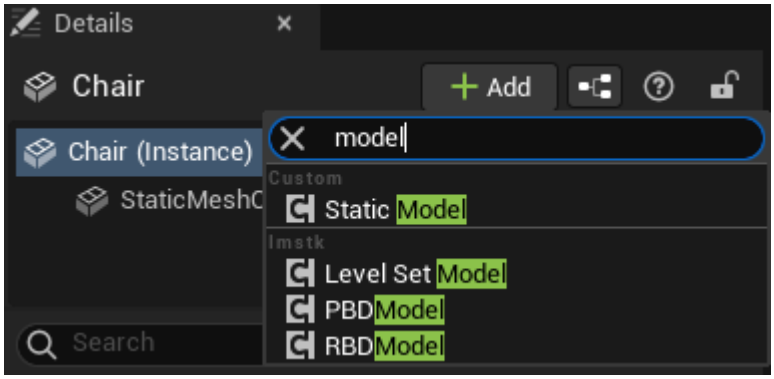
iMSTK Settings

1. Settings for IMSTK can be found under Project Settings -> Game-> iMSTK Settings

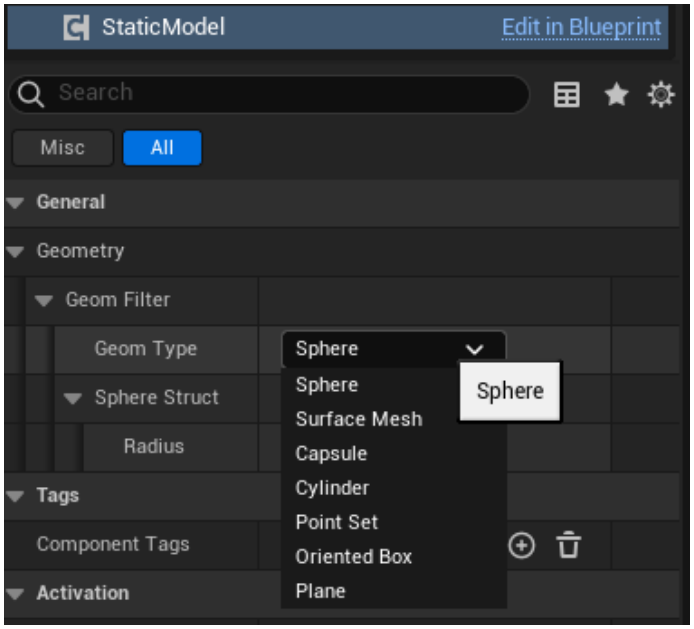
Adding Models to iMSTK

Static/RBD Models

1. Add the relevant model component to the actor



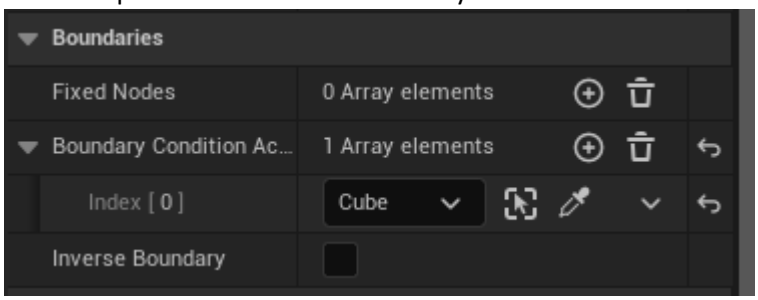
2. Under Geom Filter choose the type of geometry the object will have in iMSTK



Note: Surface Mesh and Point Set require a static mesh component to be attached to the actor. The other geometry types are generated in iMSTK without visuals in Unreal

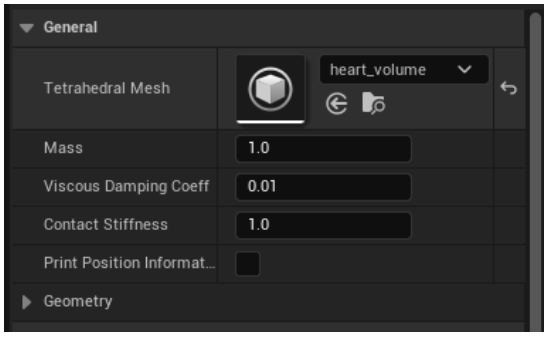
PBD Models

1. See steps for static and RBD models
2. Vertices can be explicitly stated for fixing on the PBDModel or an actor with a static mesh component can act as a boundary condition to fix all vertices within it



3. For tetrahedral meshes, import the tetrahedral model into Unreal. This will generate a Tetrahedral mesh asset and a static mesh of the model.

4. Assign the tetrahedral mesh on the PBDModel component and the static mesh on the static mesh component (The static mesh does **not** need to be the generated mesh from importing the tetrahedral model)



5. The relevant constraints can be assigned on the PBDModel

Note: The geometry on PBDModels must be set to surface meshes or point sets

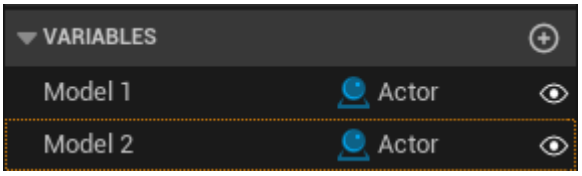
PBD Threads

1. Create an empty actor
2. Attach a spline component to the actor
3. Attach the PBDThread component to the actor

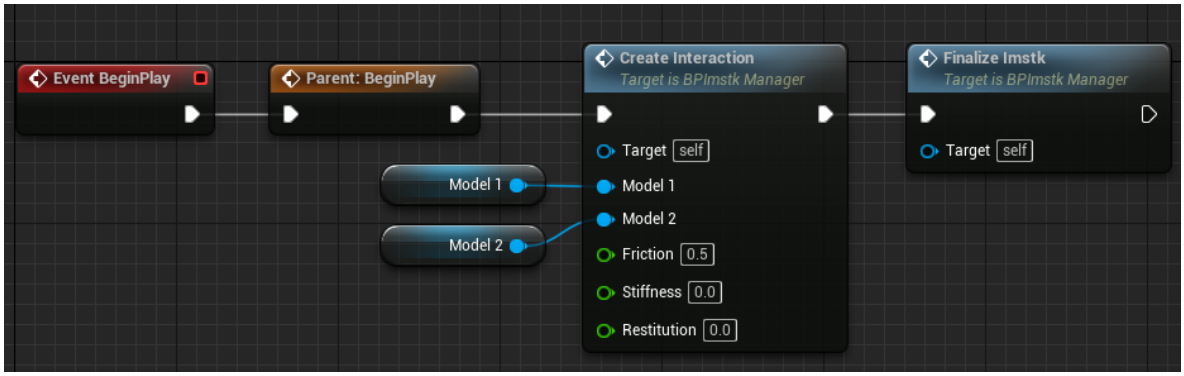
Spline meshes can be used to visualize the thread. A blueprint is included with the plugin that can be used.

Adding iMSTK Interactions

1. Open the iMSTK manager blueprint
2. Create variables of type Actor for each model interacting and set the variable to public by clicking the eye

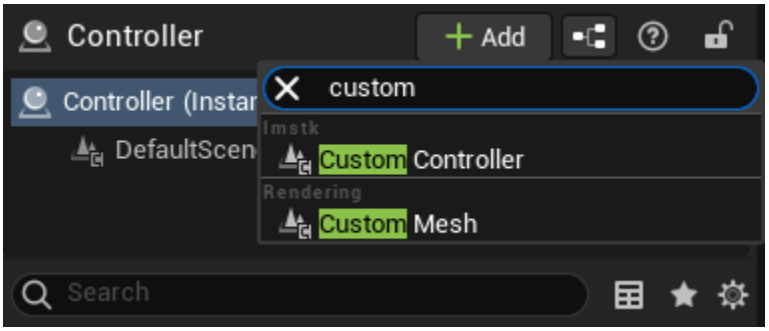


3. Assign the models on the Manager to the relevant Actors
4. Add “Create Interaction” nodes between Parent:BeginPlay and Finalize iMSTK nodes and assign parameters

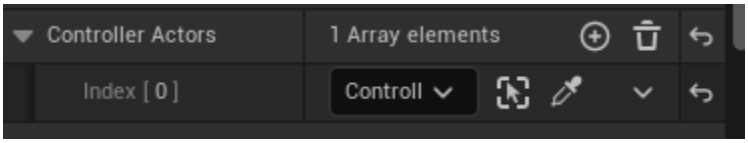


Adding iMSTK Controllers (WIP)

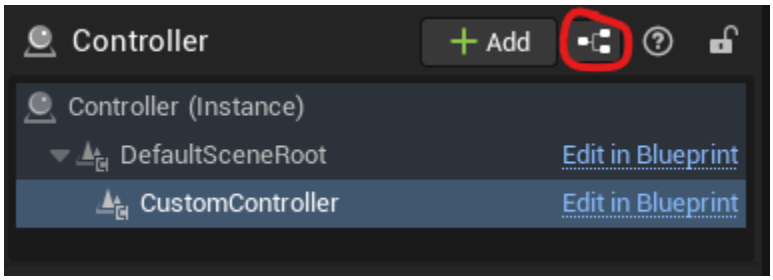
1. Attach the CustomController component to the actor



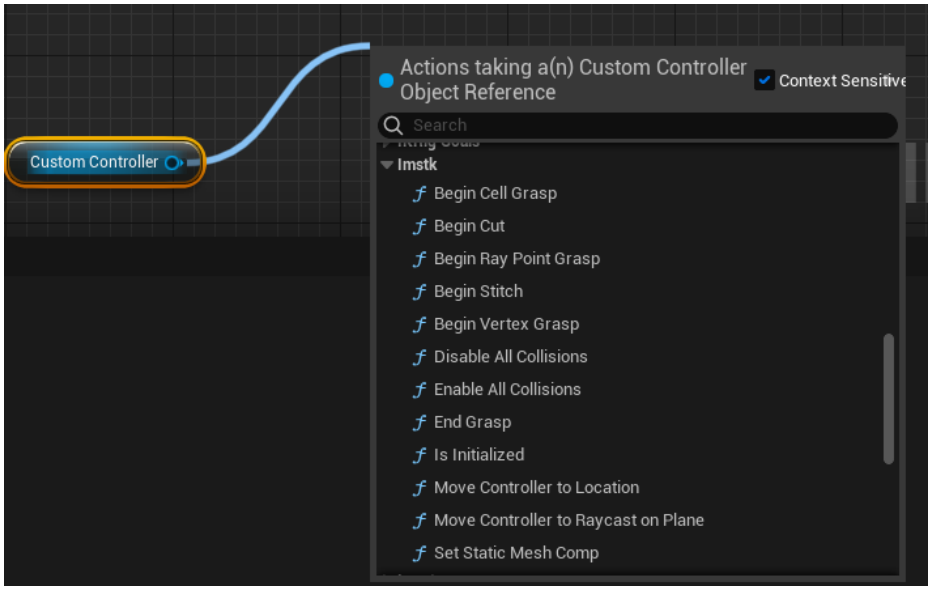
2. Set the relevant variables on the controller. Inertia tensor multiplier must be greater than 0 and the tool geometry and tool type must match (Grasping tool must be a capsule or sphere, stitching tool must be a line mesh, cutting tool must be a surface mesh and colliding tool may be any geometry)
3. Add the controller actor to the iMSTK manager blueprint



4. Create a blueprint from the actor



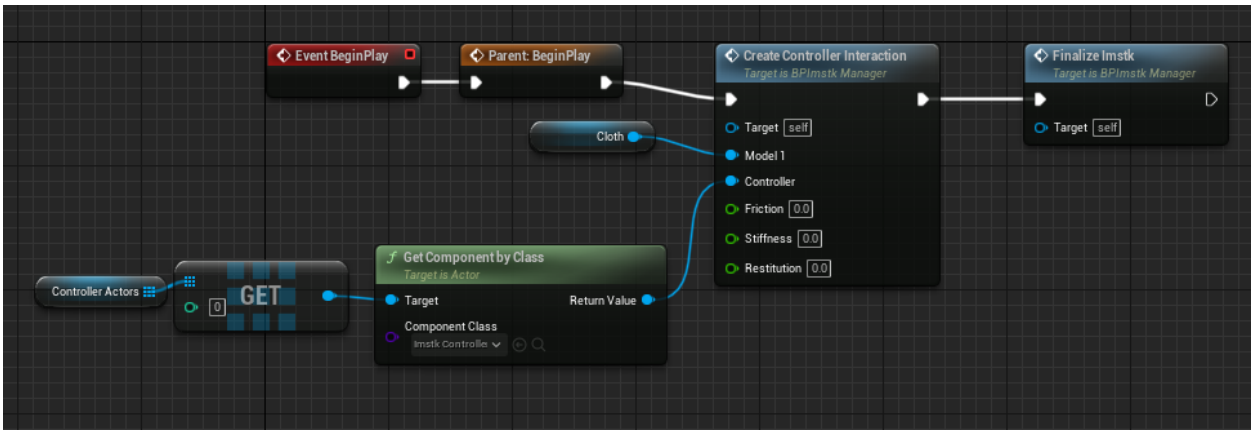
5. The controller functions can then be accessed from the blueprint



Note: For a static mesh controller SetStaticMeshComp must be used in the construction script of the blueprint to assign the variable.

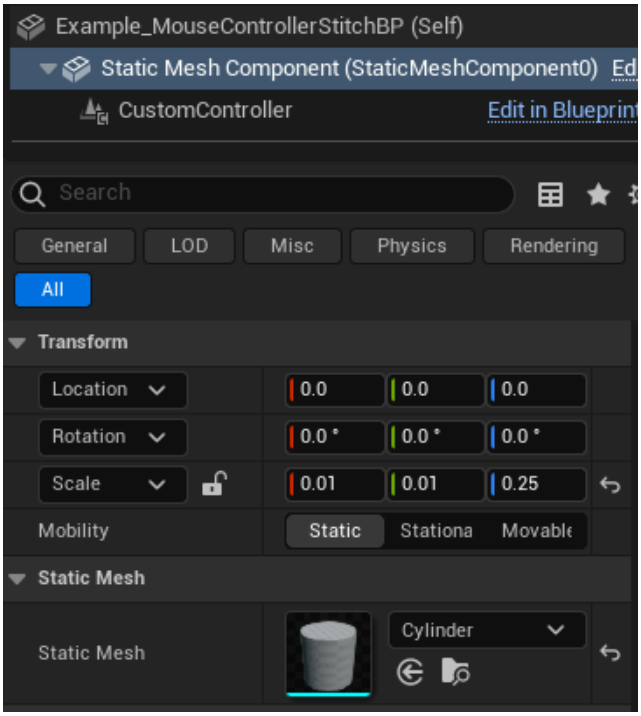
Adding Controller Interactions (WIP)

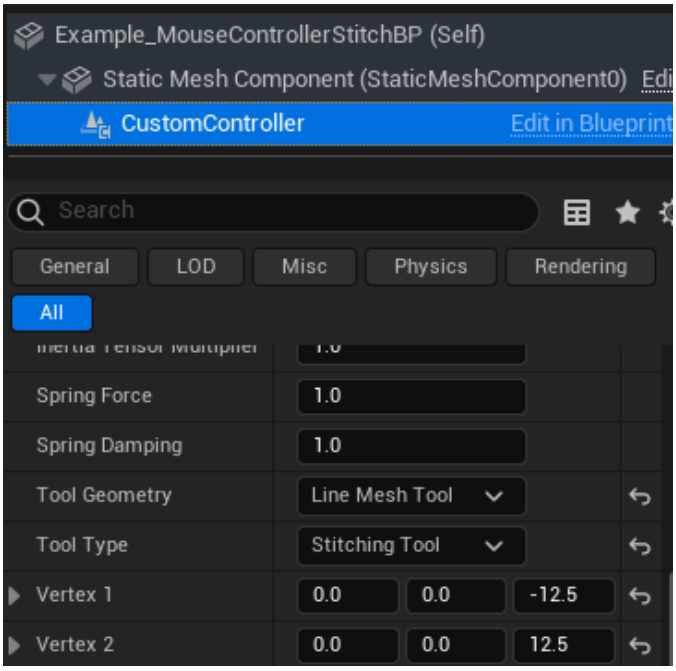
1. Open the iMSTK manager blueprint
2. Add a variable for the object you would like the controller to interact with
3. Add “CreateControllerInteraction” and assign the parameters



Example Maps

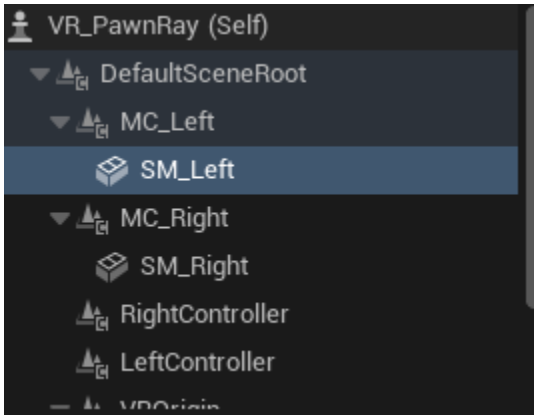
For the “Example_MouseStitching” map, the static mesh of “Example_MouseControllerStitchBP” must be set to “Cylinder” and the vertices on the line mesh for CustomController must be changed to -12.5 and 12.5.

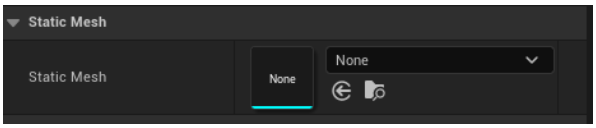




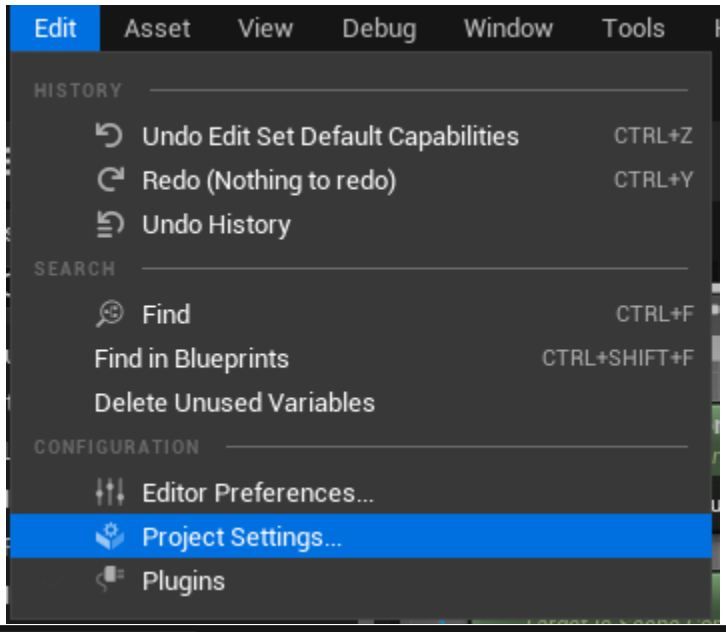
For VR examples

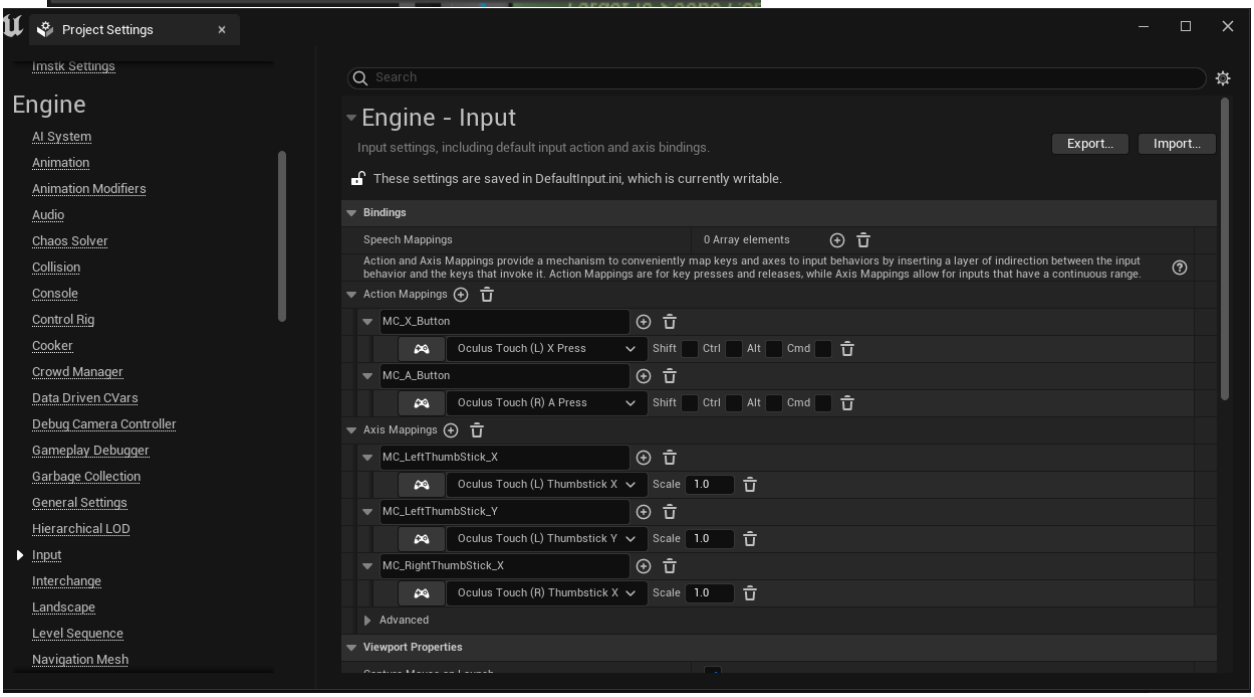
1. The static meshes must be assigned for the motion controllers on the VR_Pawn blueprint (this is just for visualization within Unreal, but the geometry assigned should be the tool geometry for that controller with the exception of VR_PawnCutting->SM_Right set to a plane for cutting).





2. The Unreal project must be set up for the headset being used for VR examples (for the Oculus Quest 2 https://developer.oculus.com/documentation/unreal/unreal-quick-start-guide-quest/)
3. The inputs for the controllers must be set in the project settings





These examples have only been tested with the Oculus Quest 2.

Example_VRThreadGrab is currently not functional.