MESHFORGE USER MANUAL

QUICK NOTE: The material in this manual discusses the splitting of skinned mesh renderers, in reality, it is splitting the mesh in the skinned mesh renderer but preserving or updating its skinned mesh attributes. For the purposes of consistency and differentiating typical static meshes from meshes found on a skinned mesh renderer the meshes that are being split/gap filled will be referred to as skinned meshes

Don't like reading? Go to the following link for video tutorials:
https://www.youtube.com/@CARRYFORGE

1. Introduction

MeshForge is a Unity Editor toolset currently consisting of five primary tools:

- **Mesh Rebaker:** Rebakes a mesh into unity's coordinate system. Used to fix mesh orientation offset issues caused by 3D modeling software coordinate systems.
- **Splitter**: Splits a skinned mesh into separate modular pieces, enabling users to reuse or swap mesh parts easily.

- **Gap Filler**: Fills holes or gaps in a skinned mesh by selecting boundary vertices or polygons and automatically generates new geometry to fill that region.
- **Skinned Mesh Combiner:** Combines skinned meshes, sharing the same bone hierarchy, into a single skinned mesh.
- **Topology Splitter:** A tool for topographically separating static meshes into split parts allowing for nuanced grouping and prefabbing.

This manual details each tool's interface, settings, and usage instructions to help you integrate **MeshForge** into your workflow. **MeshForge** is meant to bridge the gap for individuals with little to no experience in 3D modeling or animation who just want the results of more modular mesh components.

2. Installation & Requirements

1. **Unity Version**: Compatible with Unity 2021LTS or later (tested up to Unity 6).
2. **Scripting Runtime**: .NET 5.x or higher recommended.
3. **Dependencies**:
    a. The package relies on UnityEditor and standard UnityEngine libraries.
    b. The latest version of Unity Burst
    c. The latest version of Unity Collections
    d. The latest version of Unity FBX exporter
    e. The latest version of Unity Mathematics
4. **Installation:** When installing MeshForge from the package manager if you are missing any of the listed dependencies you will be met with a pop-up window. By clicking 'OK' all of the missing dependencies will be installed. **NOTE:** Give Unity some time to install the packages and ignore any errors or warnings until installation is complete.

3. Rules

3.1 Comprehensive requirements

The following is a list of rules/ guidance to make sure that your time with **MeshForge** goes without a hitch.

- **Always** place the prefabs or game objects you need to work with into the scene first then set its position and rotation to 0,0,0 this ensures a clean matrix for **MeshForge** to work with.

The rest are specific to the splitter tool when pressing the UseCustomPrefab toggle to true:

- Make sure that the prefab that is used is the **same prefab** the target skinned mesh is coming from.
- If the two skinned meshes (source and target) use vastly different bone hierarchies expect abnormal deformations. This can **possibly be fixed** by **allowing deformation** and using **pose-based deformation**. However, two skinned meshes with vastly different bone hierarchies in general will not work across one another for a number of reasons.
- To ensure that split skinned meshes have no abnormalities and deform correctly **only** share meshes across prefabs where the **same bone hierarchy** is used. This will **'typically'** be seen in assets that are made by the same artists.
- There are situations where you can share split skinned meshes across different bone hierarchies. Specifically, for instance, in cases where there are **ONLY A FEW** missing bones from the target skinned mesh that the source skinned mesh has. In these cases, one type of deformation **IS REQUIRED**.

  NOTE: 'missing bones' do not include skinned mesh assets that are placed in a bone hierarchy. Those transforms will still appear in the bone loader, but they can be ignored. I am specifically talking about **bones that have significant influence on the skinned meshes animations**.

- If either the target or the source skinned mesh was corrected using the **Mesh Rebaker** you **WILL** experience deformities. **Either both** need to have been **Rebaked FROM THE SAME OFFSET or neither rebaked** at all. To ensure there are no issues simply **only use** source skinned meshes and target skinned meshes that **share the same bone hierarchies/offsets**. There is a caveat to this rule, but every rule above must also be considered and that is by using the pose-based deformation you could end up with a functioning/properly skinned mesh.

4. Mesh Rebaker Tool

4.1 Overview

When assets are exported from 3rd party 3D software environments, they may have failed to be placed into unity's coordinate system. While this may be fixed in a number of ways by the asset creators such as adding a transform to offset the mesh orientation, this orientation offset makes assets unusable by default with **MeshForge.** So to fix this we have the **Mesh Rebaker.**

The **Mesh Rebaker Tool**:

- Takes in a single prefab but reorients all skinned meshes attached to said prefab
- Offers options to add additional offsets in case the default reorientation does not work on its own.
- Produces an entirely new prefab with the adjusted skinned meshes and reoriented bone hierarchy, creates all new reoriented meshes, and creates a new FBX so that a new humanoid avatar can be made for the new prefab.
- The original asset is not changed and can still be used elsewhere

4.2 Interface and Settings

Open via **Tools → MeshForge → One Shots → Mesh Rebaker**.

Below are the primary fields and options in the **Mesh Rebaker Window**:

1. **Save Folder** (Required)
   a. Sets the subfolder path under `Assets/` where the output meshes and prefabs will be saved.
   b. Example: If you enter `MyGame/Meshes`, the final path becomes `Assets/MyGame/Meshes`.

2. **Source Prefab** (Required field)

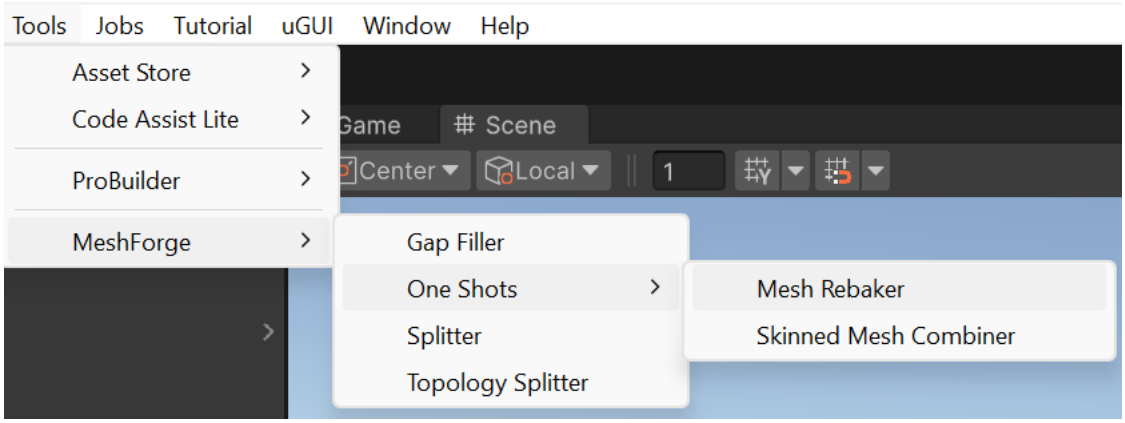
a. Drag your prefab from the scene and place it into the field

b. This represents the prefab that will be reoriented

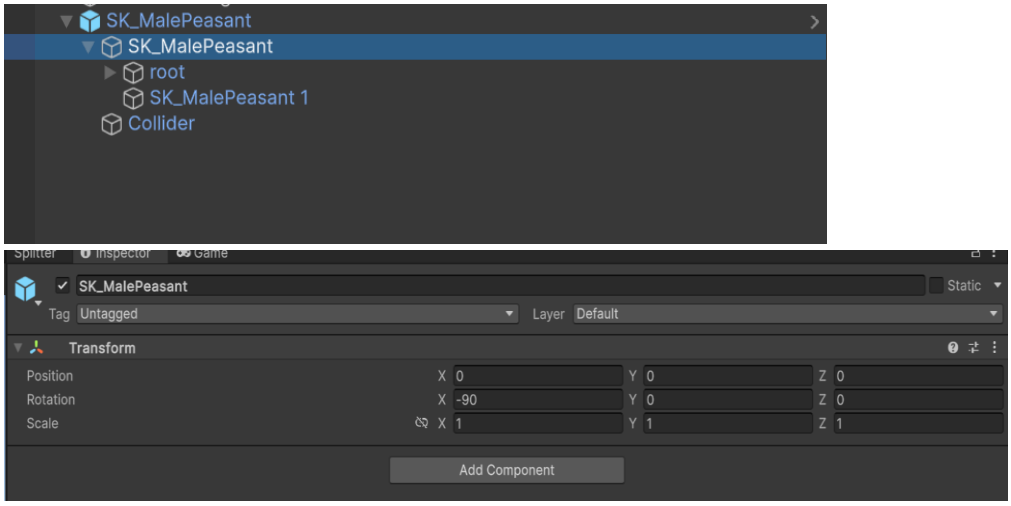
3. **Offsets fields** (Optional)
    a. These fields are used to provide an option for further adjusting the offset of the mesh if necessary.
4. **Rebake Meshes and Create Assets** (Button)
    a. Press this button to finalize the reorientation

4.3 Mesh Rebaker Quick Start

1. **Open Mesh Rebaker window**
        a. **Tools → MeshForge → One Shots → Mesh Rebaker**



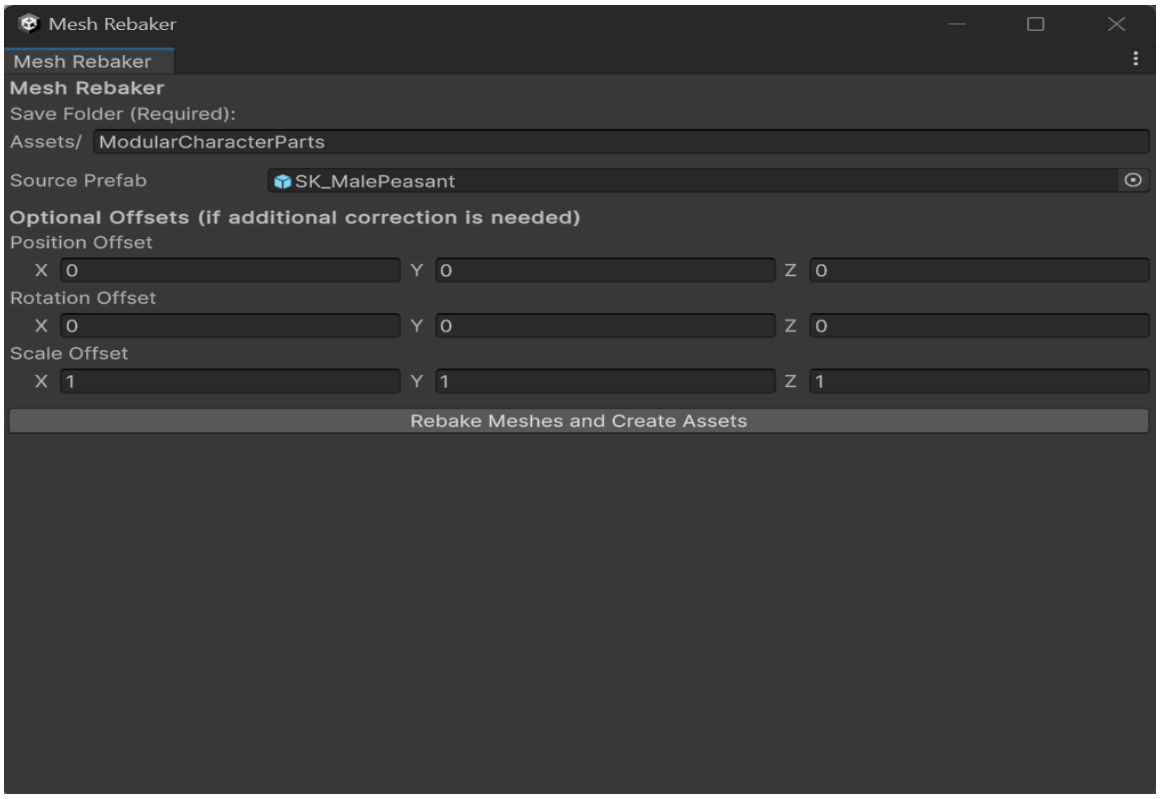
2. **Drag and drop the prefab you want to reorient into the scene/ confirm it has an abnormal rotation.**
    a. You can see a mesh needs reorientation a few ways, I will show two. First when placing a prefab into the scene look at any transforms that are parents to the bone hierarchy you will see an offset on the parent, in our case a –90 offset on the x axis:



b. You can also just go into the asset folder and look at the mesh directly. You can see here that the mesh is oriented abnormally from unity's standards. The mesh is laying flat treating the +z axis as the +y in terms of normal mesh orientation.

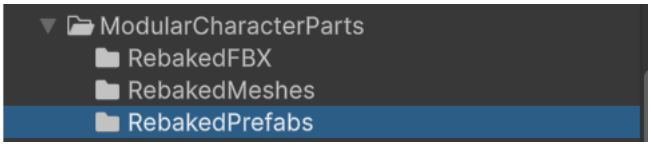
3. **Place the prefab from the scene into the prefab field**
4. **Apply any optional offsets.**
   a. For our mesh we do not need to add any additional offsets

5. **Press the ‘Rebake Meshes and Create Assets’ button**



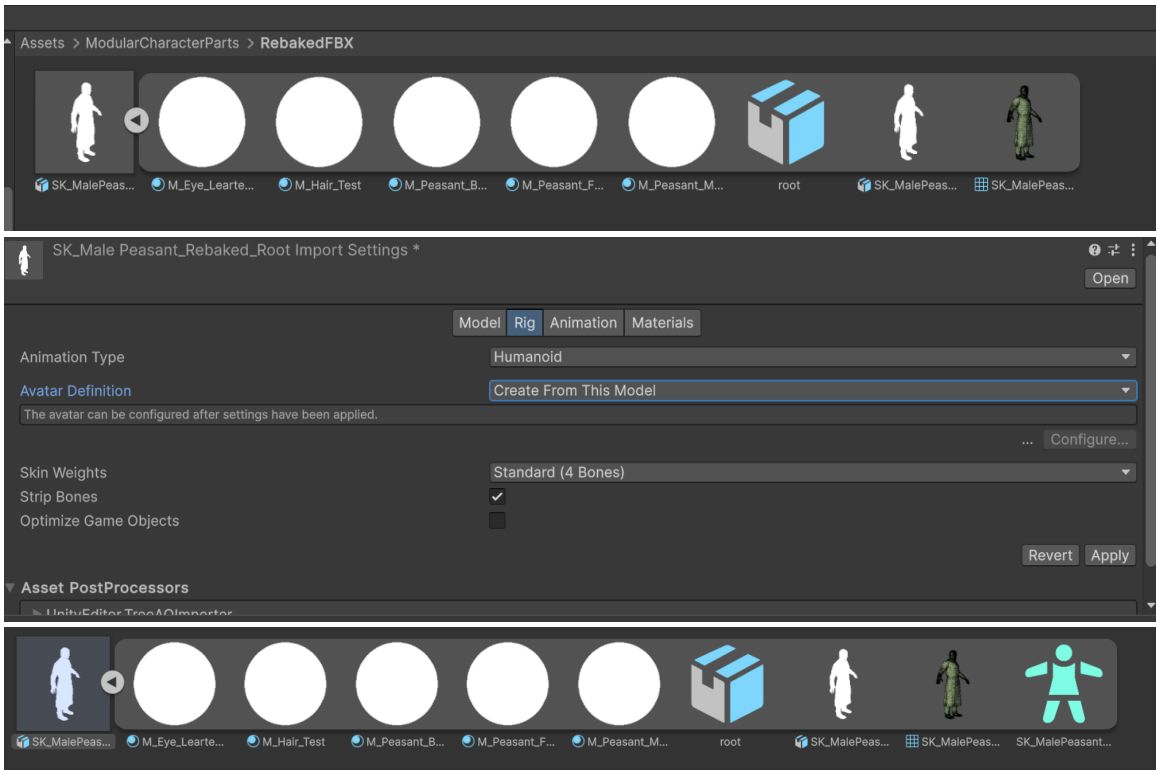
6. You will now see three new folders containing your newly created assets.

a. Rebake as many times as you like with any number of offsets until you get the results you want. The rebake functionality will simply replace/overwrite your previous rebake results so you won't have cluttered folders.



7. Go to your fbx and create a new humanoid avatar for your new prefab.

a. Click the fbx file, go to the rig tab in the inspector, choose Humanoid for the animation type, choose Create from this model for the avatar definition, and click apply. You will now have a new character avatar.



5. Splitter Tool

5.1 Overview

Modular characters often require separating a single skinned mesh into multiple parts while preserving attributes for things like animations. For instance, you might want the character's arms, legs, and torso as separate swappable pieces.

The **Splitter Tool**:

- Split a single *Skinned Mesh* into multiple Skinned Meshes by groups.
- Allows advanced *flood fill*–type selection of which triangles go to which group.
- Preserves the original attributes or updates them, so each piece remains skinned/ animatable.

5.2 Interface and Settings

Open via **Tools → MeshForge → Splitter**.

Below are the primary fields and options in the **SplitterWindow**:

1. **Skinned Mesh/ Static Mesh (tabs)**
    a. Allows the user to switch between skinned mesh and static mesh versions of the gap filler window.

1A. Static Mesh Tab Specific Fields (the following can be found in the settings foldout of the group management box)

a. **Overwrite Saved Objects** (toggle):
    i. An optional toggle that allows you to overwrite the object of the same name when finalizing.
    ii. Useful when you are continuously iterating changes to the mesh and want to prevent clutter in your folders.
    iii. This will also overwrite the selected skinned mesh that is attached to the game object the selected skinned mesh comes from.
b. Add Physics Components (toggle):
    i. If on it adds physics components (rigidbody /mesh collider) to each individual part. Mesh Collider is used as it is the most accurate when using colliders for separated parts.

2. **Reset** button
    a. Resets/clears all fields in the splitter window and removes any preview objects in the scene.
3. **Save Folder** (Required)
    a. Sets the subfolder path under `Assets/` where the output meshes and prefabs will be saved.
    b. Example: If you enter `MyGame/Meshes`, the final path becomes `Assets/MyGame/Meshes`.
4. **Source Skinned Mesh** (Required)
    a. The mesh you want to separate. Must have valid bones, non-empty materials, etc.
5. **Bone Weight Group Config** (Required to be set/using it is optional) (available in version 1.1.0)
    a. A field containing the bone weight config scriptable object.

6. **Ambiguity Threshold** (Optional/ available if bone weight config is not null) (available in version 1.1.0)

a. Used to define a threshold for setting boundaries when using the bone weight-based grouping.
b. While not completely reliable a value of .5 will help even out the boundaries when using bone weight-based grouping. This will make the boundaries more even/ uniform as opposed to a more jagged boundary.

7. **Use Custom Prefab?** (Optional)
   a. If checked, you will need to specify a target skinned mesh (comes from the custom prefab). Must have valid bones, non-empty materials, etc.
   b. If checked, you will need to specify a prefab to use as the home for the split Skinned Meshes.
   c. If checked, you will be given a field to input the name of the root bone from the custom prefab

8. **Target Skinned Mesh** (Required / available if 'use custom prefab?' Is set to true)
   a. allows you to match or retarget your source skinned mesh to a new bone structure or allow deformation.

9. **Custom Prefab** (Required / available if 'use custom prefab?' Is set to true)
   a. Prefab will come directly from the assets folder. The prefab should contain the following: One full character(Non-split) SkinnedMeshRenderer, a valid bone hierarchy.

10. **Prefab Root Bone Name** (Required/ available if 'use custom prefab?' Is set to true)
    a. This should be consistent with the name of the root bone in your custom prefab. The root bone is typically the first bone of the bone hierarchy. However, if you are unsure you can go to the SkinnedMeshRenderer on your prefab go to the root bone field this will show you the name of the root bone. If you click on the transform in the root bone field, it will show the location of the root bone.

11. **Allow Deformation** (Optional/ (experimental) available if 'use custom prefab?' Is set to true)
    a. If set to true, your source's mesh is deformed to match the target mesh's pose or using a lattice-based approach.

12. **Use Pose Based Deformation?**(optional/ available if ‘Allow Deformation?’ is set to true)
    a. If marked as true during the deformation step it will deform using bind poses. (Best for neutral to neutral size meshes or going from smaller meshes to larger target meshes).
    b. If marked as false during the deformation step it will deform using a lattice based deformation. (Best for neutral to neutral size meshes or going from larger to smaller target meshes.)
13. **Lattice Resolution** (optional/ available if ‘Use Pose Based Deformation’ is set to false)
    a. Used to define the resolution of the output mesh during deformation. Resolution is key to the quality of the deformation 300 appears to be a good resolution for low poly meshes without any visual loss. **WARNING: placing the resolution too high will crash unity. Due to the differences in every individual’s hardware, it is up to the user to determine what “too high” means.**
14. **Hidden Material** (Required)
    a. Used to “hide” submeshes in the editor while you isolate certain triangles.

15. **Use render pipeline default material for mesh preview?** (optional)
    a. If marked as true, the render pipelines default material will be used for the mesh preview. If marked as false, then the source skinned meshes materials will be used for the preview

16. **Load Bones** button (Required/ available if ‘use custom prefab?’ Is set to true)
    a. Initializes the Bone Hierarchy Sub-Editor.
    b. Displays two lists. The first is the source skinned mesh bones the second is the target (custom) prefab bones.
    c. Allows you to ensure that any bones that should behave like one another can have matching names. This is important as this is how the splitter determines how the bones should be retargeted from the source skinned mesh to the target prefab bone hierarchy.
    d. The ‘Apply Names’ button overwrites the names that were changed and then initializes the ‘Load Mesh Data’ button.

e. Example: if on the source you had Hand_R and on the target you had Right_Hand you could change the name on either the target or the source to fit the naming convention that you prefer. **NOTE: changing the name on the source skinned mesh is the best option.**

17. **Load Mesh Data** button
    a. Reads the mesh data from the Source Skinned Mesh.
    b. Creates an in-Editor preview object (a standard MeshFilter / MeshRenderer) for you to select triangles from.

    This is true for flood filling as well. If the submesh in the submesh visibility settings is not selected and the flood fill occurs those 'hidden' triangles will not be selected during the fill.

18. **Group Management**
    a. **Add New Group** button: Adds a new group to the group management section as a foldout. Any newly added group becomes the 'Active Group'
    b. **Reset All Groups** button: Resets all Groups to their default states (does not remove groups)
    c. Advanced Grouping button (available in version 1.1.0): uses the bone weight-based grouping algorithm and the bone weight grouping config file to create groups.
19. **Settings** (foldout)
    a. **Submesh Visibility**(toggle list)
        i. If toggled on that submesh will be visible/paintable. If toggled off, then it will not be.
    b. Flood Fill Settings
        i. **Proximity Threshold**(Slider): Controls adjacency rules for triangles (how strictly edges must match).
        ii. **Seed Direction Threshold**(Slider): Used for directional flood fill. If a new triangle's direction matches the boundary's direction, it belongs to the group.
    c. Unprocessed Triangle Edges Settings

i. **Show Unprocessed Triangles**(toggle): Turns the views of the triangles that have not been processed (selected) yet on or off.

ii. **Unprocessed Edge Color**(Color selector): Determines the color of the unprocessed triangles.

iii. **Handle Draw Distance**(Slider): Controls how far (in world space) from the SceneView camera a triangle handle is visible. Triangles beyond this distance are not shown.

iv. **Triangle handle z offset**(slider): Sets the distance away from the meshes surface in the forward direction the triangle will be drawn. The higher the value the further away from the surface of the mesh each triangle will be drawn.

d. Finalize Settings

i. **Allow Ungrouped triangles to be made into their own group/mesh**(toggle): when selected any triangles not in a group will be placed into their own group and included in the final skinned mesh prefab/updated prefab.

20. Groups

a. MODES: Each group contains three selection modes.

1) **Boundary Mode:** When triangles are selected in this mode triangles will act as boundaries. Boundaries can be thought of as any enclosing shape such as a circle/square/cylinder where it encloses a group of non-selected triangles. **Boundary Mode** works directly with **Seed Mode** to prepare the mesh for flood filling. The reason the shape needs to be enclosing is that when the flood fill occurs the flooding will begin at the boundary and work towards the seeds.
2) **Paint Mode:** This mode allows the user to select triangles freely with no direct connection to the initial flood fill. This means that any triangles set in **Paint Mode** will not be used as either a boundary or a seed. It is useful for selecting triangles that the flood fill didn't reach or for a more customized mesh separation approach.
3) **Seed Mode**: Contrary to the name when in seed mode the seed will act as a guide for the flood fill. The triangles flood from the boundary triangles to the seed triangles. There is no limit to how many seeds

you can place and ideally you should place as many as you think necessary so the flood fill can reach all areas of the mesh. This is especially true for skinned meshes with very complex geometry (complex = many peaks and valleys/divots).

NOTE: When using boundary and seed mode the triangles set in those modes will not actually be applied to the final Skinned Mesh in their respective groups unless the group is "processed", use paint mode if you want to freely add triangles to a group without the need to process the group.

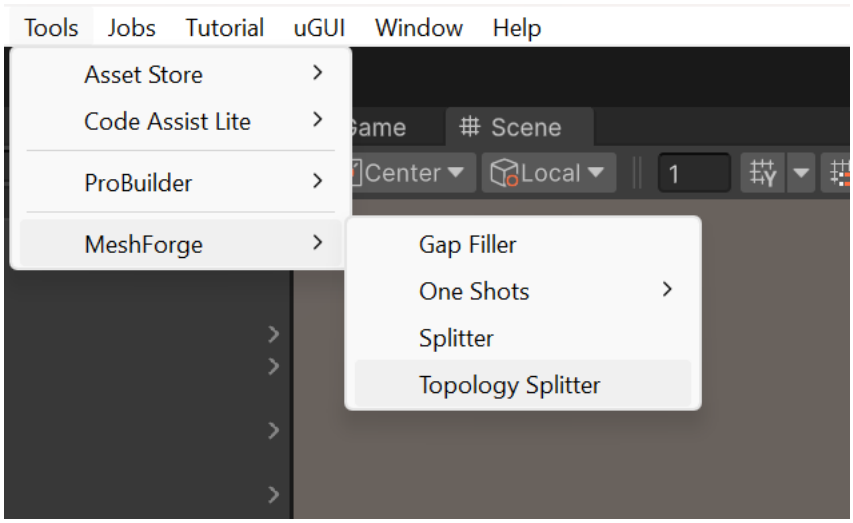
b. **Group Name** field: Use this field to define the name not only of the group but eventually this name is what will be used for the name of the skinned mesh part. Its best to name it as the body part it represents. **NOTE**: **come up with a naming convention and stick to it so that you can stay organized.**
c. **Group Color** field: Allows you to quickly identify what group you are working with.
d. **Process Group** button: Begins the flood fill process for this specific group.
e. **Process Incomplete Group** button (only available after process group has been used once): Begins a secondary flood fill for this specific group. It uses all the currently processed triangles of this group to try and fill in any neighbors that are still not processed while respecting the original boundaries.
f. **Remove Group** button: removes the active group from the group list.
g. **Prev State** button: If available it will return the group to its previous state.
h. **Next State** button: if available it will move the group to its next state.

21. **Separate Part(s) and Save** (button)/(only available when one group has been added to the group list)
    a. Split the skinned mesh into separate skinned mesh parts based on your group selections.

5.3 Splitter Quick Start

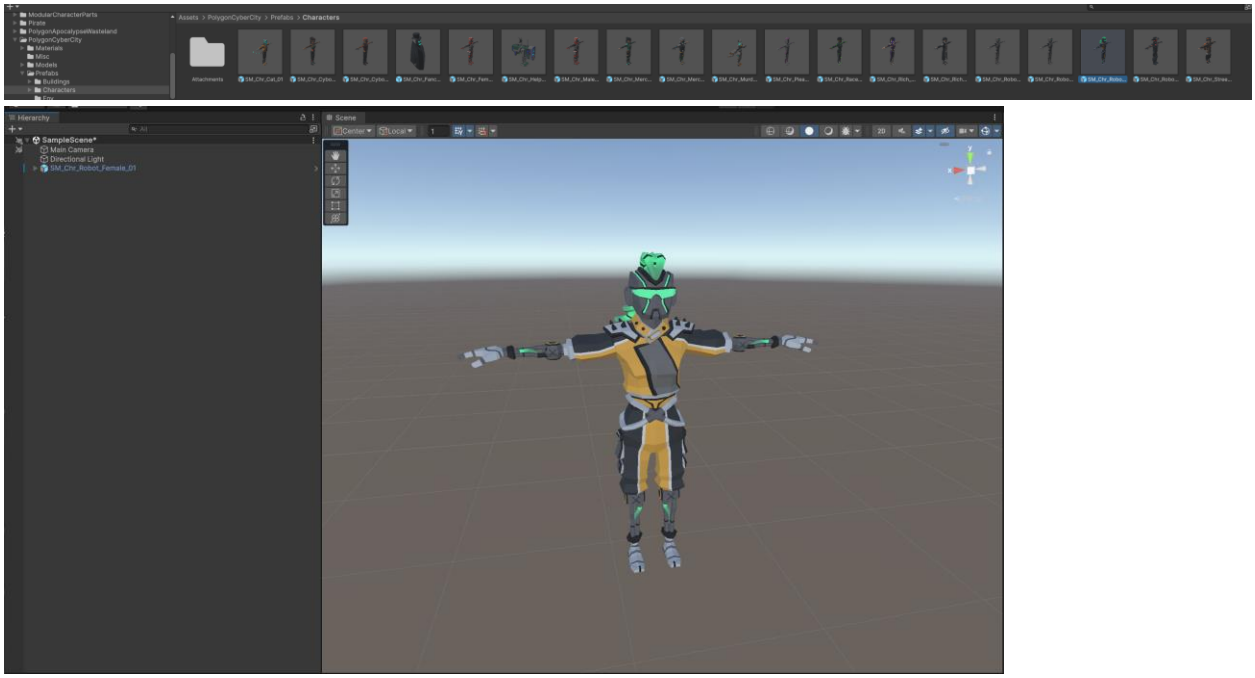
1. Open splitter window

a. **Tools → MeshForge → Splitter**



2. Drop Gameobject into scene

a. Drag and drop a prefab or gameobject asset from the asset folder to the scene hierarchy.

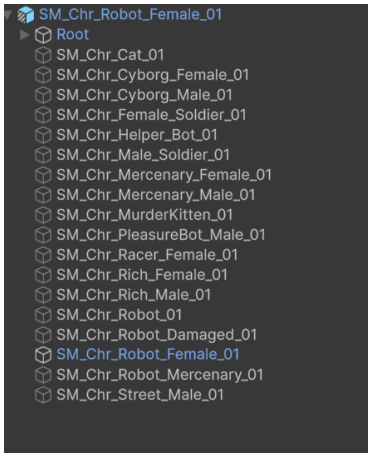


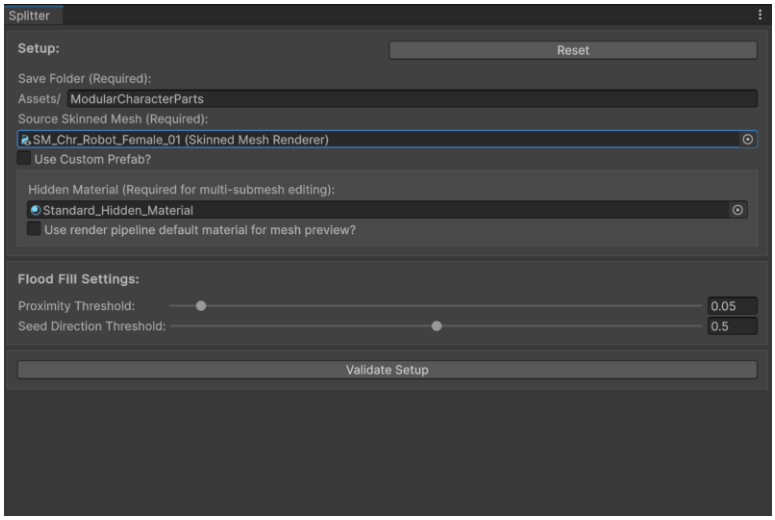
3. Leave all fields at their defaults

a. In the splitter window do not change any of the default values for brevity.

4. Assign Source Skinned Mesh

a. Pick the skinned mesh from the prefab in the scene you want to split. And drag it into the source field of the splitter window





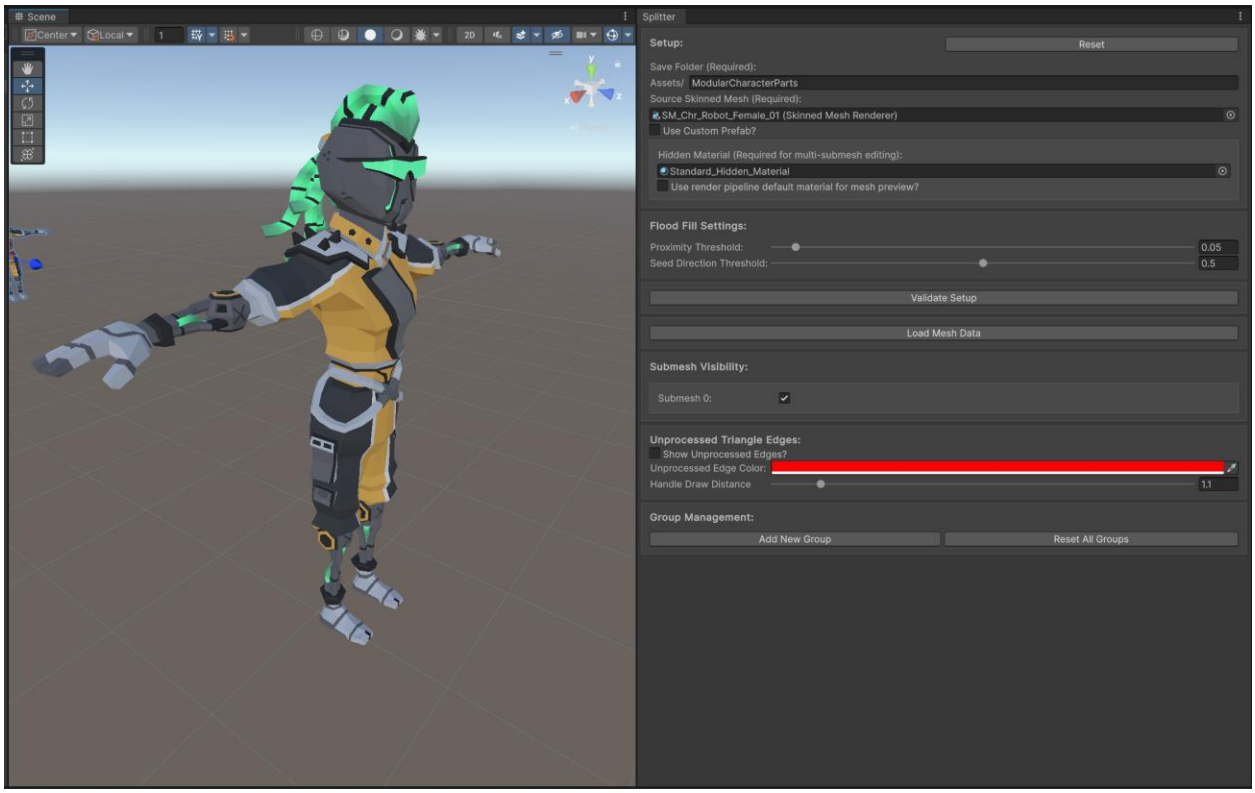
5. Click “Validate Setup”

a. Here if their are any issues with your skinned mesh they will be displayed at the bottom of the ‘Splitter window’.

6. Click “Load Mesh Data.”

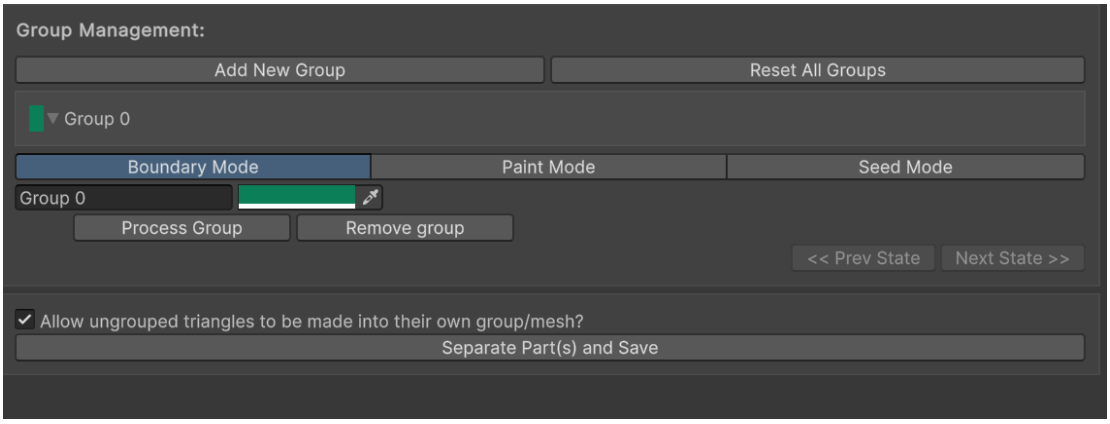
a. A preview object is created in your Scene.

b. Submesh toggles let you show/hide submeshes.



7. Click “Add New Group”

a. A new group will be added to the ‘Splitter Window’.



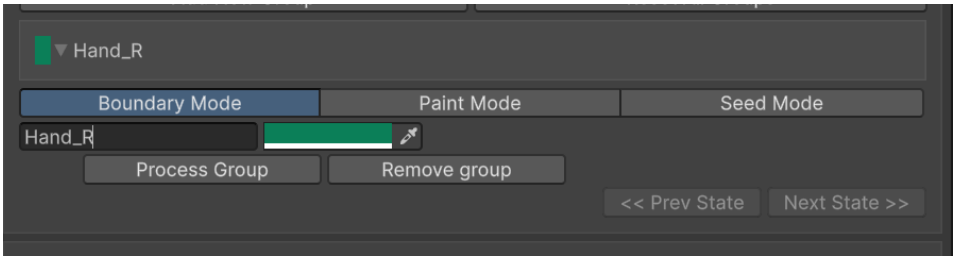
8. Rename Group

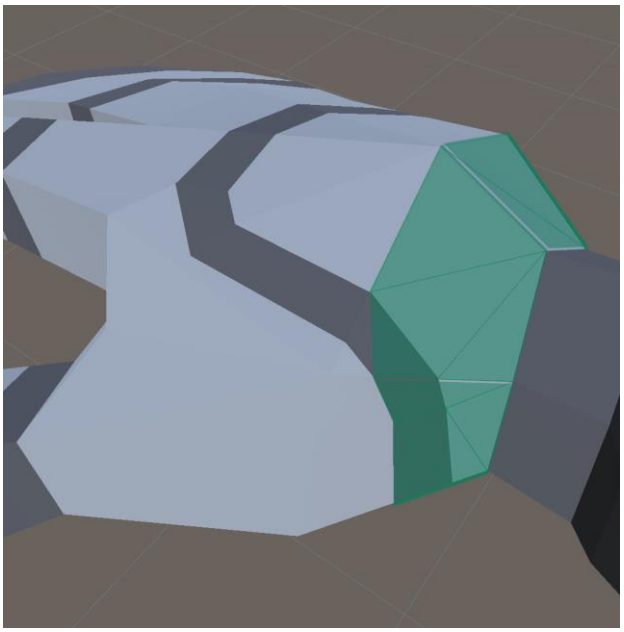
a. Each group can represent a body part or region. For example, “Hand_L,” “UpperArm_R,” “Torso,” etc.

9. Select Triangles

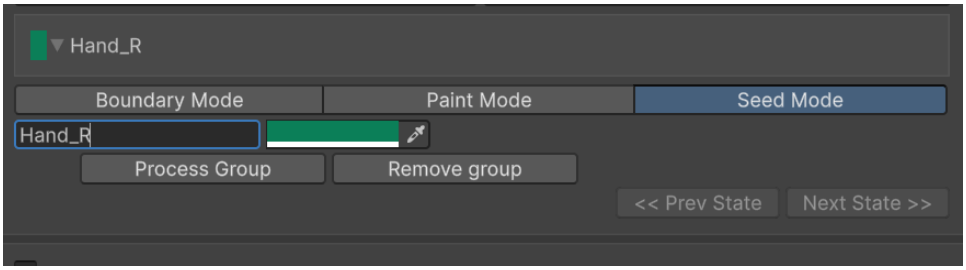
a. While in ‘Boundary Mode’ in the active group, In the Scene View, click ‘triangles’ (the mesh) to mark them for the active group. Any triangles that are marked as boundaries are inclusive meaning during the separation process, they will be included in the group that they are colored as. Make sure when creating the boundary, the boundary triangles are a ‘closed’ shape in the case of the wrist, the boundary is enclosed like a band around the entire wrist.

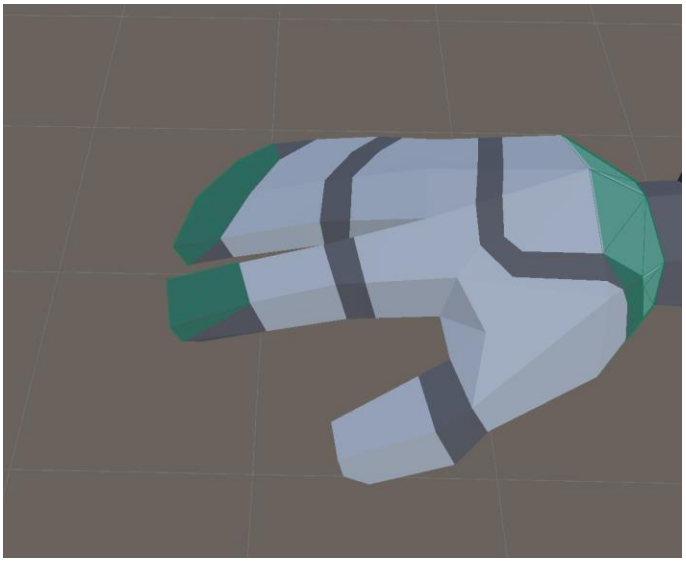
(left-click = add, right-click = remove **OR** hold left-click + drag-mouse = ‘Drag Add’, hold right-click + drag-mouse = ‘Drag Remove’).



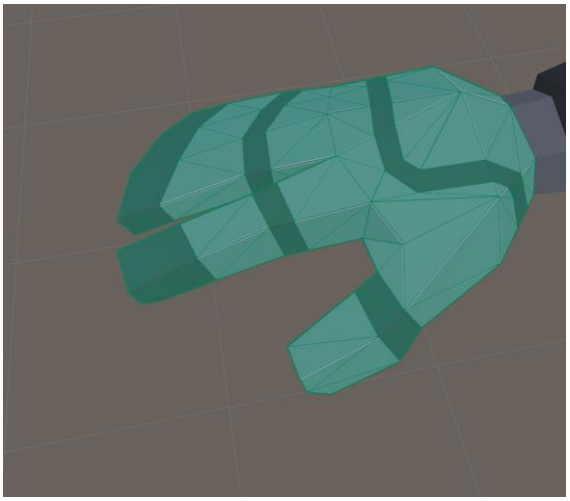


b. Change the active groups mode to ‘Seed Mode’ to begin painting seeds on the mesh. There is no limit to the number of seeds you should/ can use. Seeds are most effective when placed in hard-to-reach areas of the mesh meaning areas with complex mesh geometry and when placed opposite end of the mesh group from the boundary triangles.



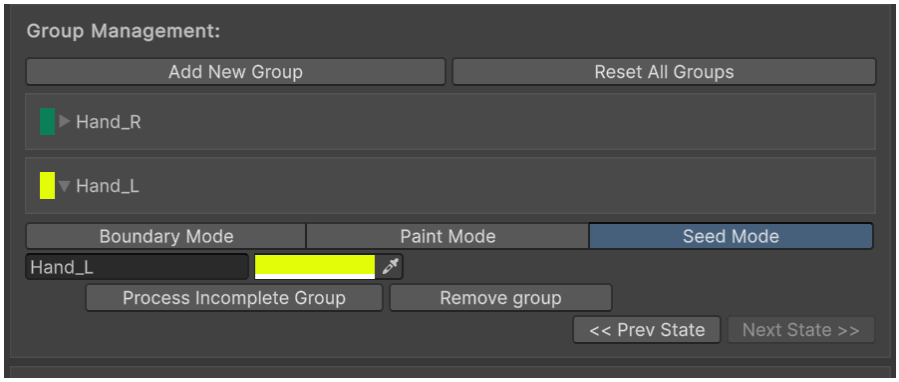


c. Press “Process Group” for a fill operation.



d. You can also use “Paint” mode to manually pick all triangles you want in a group without boundary logic.

10. Repeat steps 7-9 for More Groups

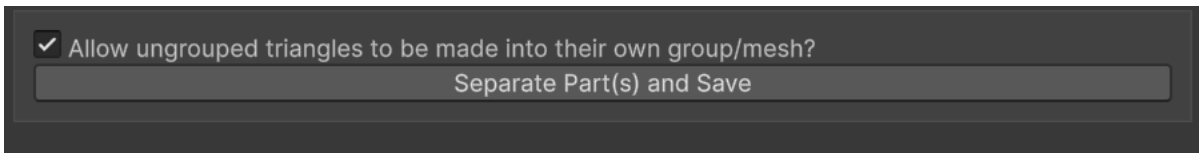


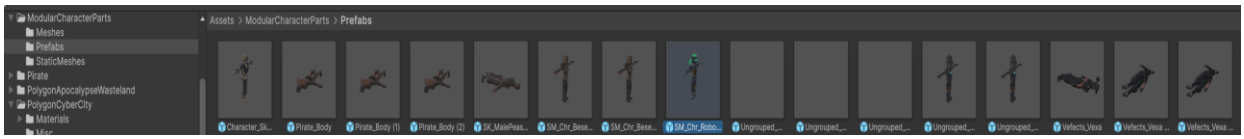
11. Separate and Save

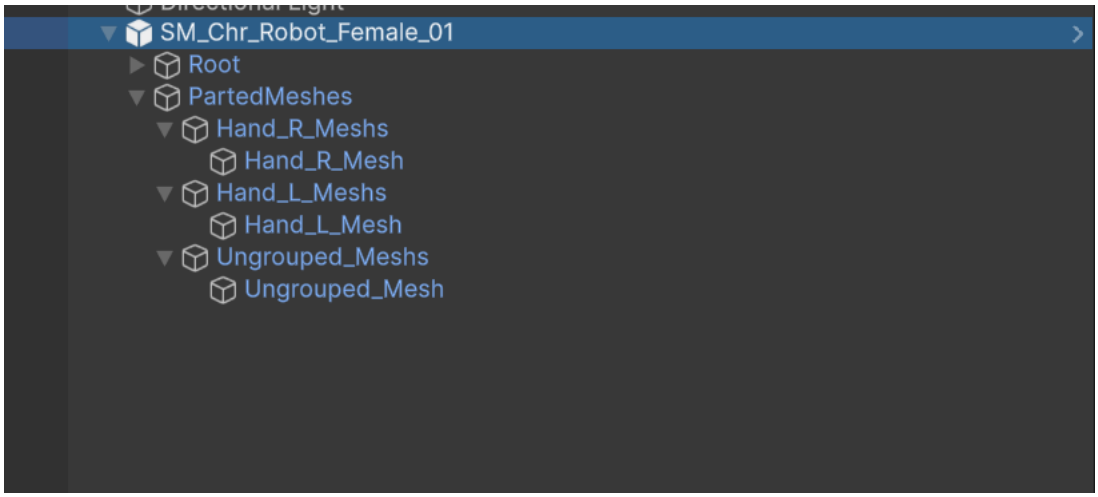
a. The tool will generate separate Skinned Mesh Renderers (one per group) and save the meshes to your chosen folder under **/Meshes**.

b. A prefab with the newly created skinned meshes and a bone hierarchy will be created and saved in your chosen folder under **/Prefabs**.

c. If you drag and drop the new prefab into the scene you will see that the skinned meshes are organized by the groups names and any skinned meshes that did not have a group are placed under a container transform called ‘Ungrouped_Meshs’.







6. Gap Filler Tool

6.1 Overview

When working with split Skinned Meshes, you will end up with holes or incomplete geometry that needs to be filled. The **Gap Filler** tool allows you to:

- Select a **Skinned Mesh Renderer**.
- Mark boundary vertices around the hole.
- Automatically fill the gap with polygons, optionally assigning a separate fill material.

6.2 Interface and Settings

You can open the Gap Filler window via **Tools → MeshForge → Gap Filler** in the Unity menu.

Below is a breakdown of each setting/option in the **GapFillerWindow**:

1. **Skinned Mesh/ Static Mesh (tabs)**
    a. Allows the user to switch between skinned mesh and static mesh versions of the gap filler window.
2. **Reset**(button)

a. When pressed the window, any preview objects, and all the fields are reset providing a clean window to work from.

3. **Save Folder (Required)**
    a. A text field labeled “Save Folder (Required):”
    b. Sets the subfolder path under `Assets/` where the output meshes and prefabs will be saved.
    c. Example: If you enter `MyGame/Meshes`, the final path becomes `Assets/MyGame/Meshes`.
4. **Settings** (foldout)
    a. **Overwrite Saved Objects** (toggle):
        i. An optional toggle that allows you to overwrite the object of the same name when finalizing.
        ii. Useful when you are continuously iterating changes to the mesh and want to prevent clutter in your folders.
        iii. This will also overwrite the selected skinned mesh that is attached to the game object the selected skinned mesh comes from.
    b. Add Physics Components (toggle)
        i. If on it adds physics components (rigidbody /mesh collider) to each individual part. Mesh Collider is used as it is the most accurate when using colliders for separated parts.
    c. Handle Draw Distance (slider)
        i. Controls how far (in world space) from the SceneView camera a vertex handle is visible.
        ii. Vertices beyond this distance are not shown as selectable handles.
    d. Vertex Handle Size (slider)
        i. Controls the size (radius) of the handle drawn on the selectable vertices.
    e. **Group Automation**
        i. **Automate Groups** (toggle): By setting this to true you will be met with new options specific to the group automation functionality.
        ii. **Contains all of the options found in both fan and fill mode** e.g. FillMode, Secondary Material, if Convex Fill Mode then: Convex Steps, Convex Exponent, Apex Offset
        iii. **Create Groups** (button): Click this to automate group creation/ filling.
5. **Gap Groups**

a. You can create multiple “gap groups,” each representing a different hole or boundary region on the mesh.
b. Each group has fields for:
    i. **Fill Mode** (`Fan` or `Convex`):
        1. Fan: Connects all boundary vertices to a single centroid.
        2. Convex: Generates convex geometry with optional “Convex Steps” for more controlled fill arcs.
    ii. **Convex Steps** & **Convex Exponent** (for `FillMode.Convex`):
        1. These control how many intermediate loops are generated and how strongly they interpolate toward the centroid.
    iii. **Apex Offset** (for `FillMode.Convex`):
        1. Distance from the boundary centroid to the apex along the computed normal. Allows the fill to “bump out” or “sink in.”
    iv. **Group Color**:
        1. The color to highlight that group’s selected vertices/triangles in the SceneView.
    v. **Gap-Fill Material (optional)**: If assigned, the filled hole polygons use this secondary material as a separate submesh.
    vi. **UV Flip Direction**: Allows rotating or flipping the UVs (options: `None`, `Rotate90`, `Rotate180`, `Rotate270`).

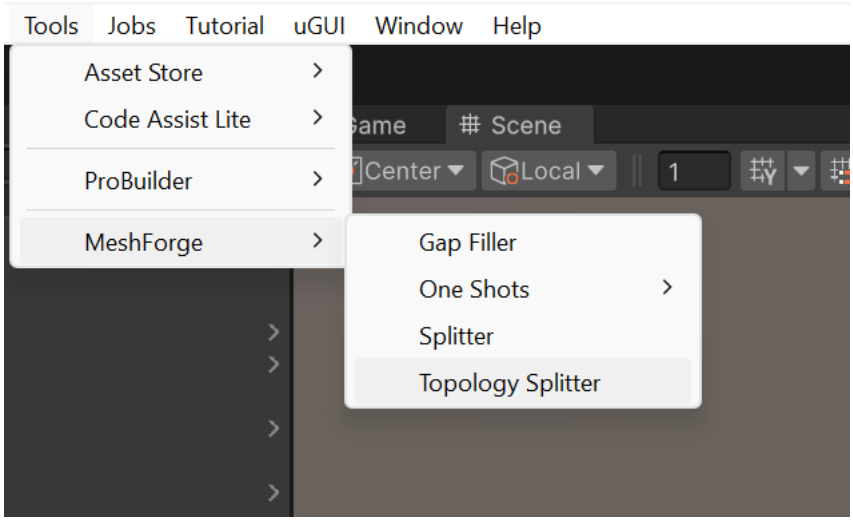
    vii. **Clear Group Selection**:
        1. Clears all assigned boundary vertices from this group.
    viii. **Remove Group**:
        1. Deletes the group altogether.

6. **Scene View Interaction**
    a. In the SceneView, each vertex (within the `Handle Draw Distance`) displays a small handle/gizmo.
    b. Clicking a handle toggles its membership in the currently “expanded” gap group.
7. **Buttons: “Update Preview” and “Finalize”**
    a. **Update Preview**:
        i. Generates a temporary fill mesh for the active group. Updates are shown on a preview object in the scene.
    b. **Finalize**:

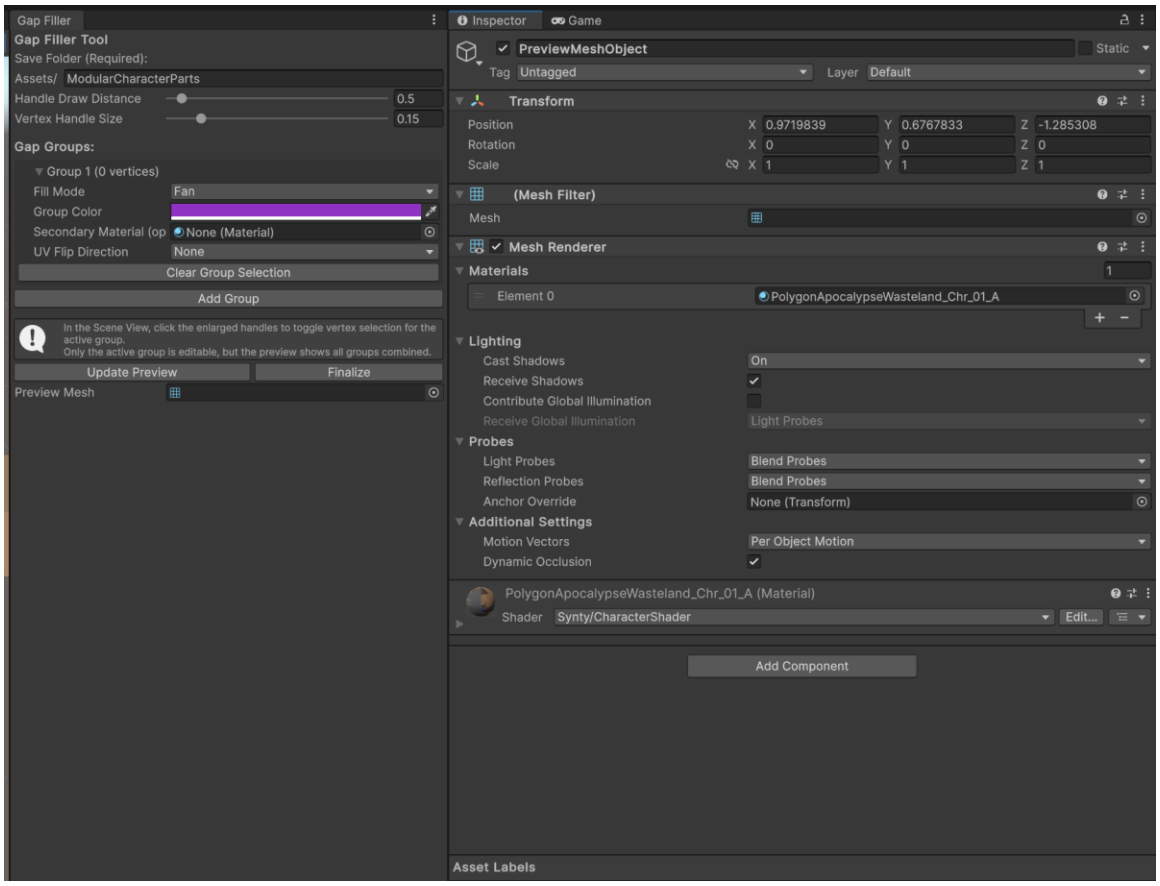
i. Commits all group fills, creating a new Skinned Mesh Renderer with the filled geometry, then saves the mesh asset and optional prefab.

6.3 Gap Filler Quick Start

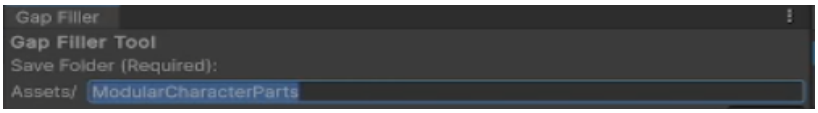
1. **Open Tools → MeshForge → Gap Filler**.



2. **Select a gameobject** in the scene hierarchy containing a *SkinnedMeshRenderer component*. **Note:** the game object you select should be located on a prefab or on a gameobject you plan to prefab.



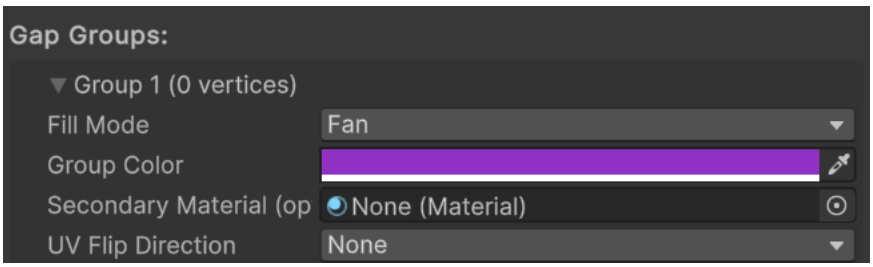
3. In the Gap Filler window:
    a. **Set the Save Folder** path if needed.

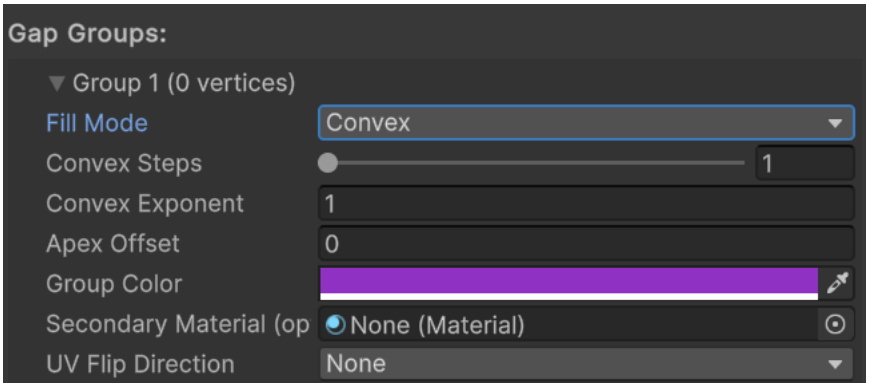


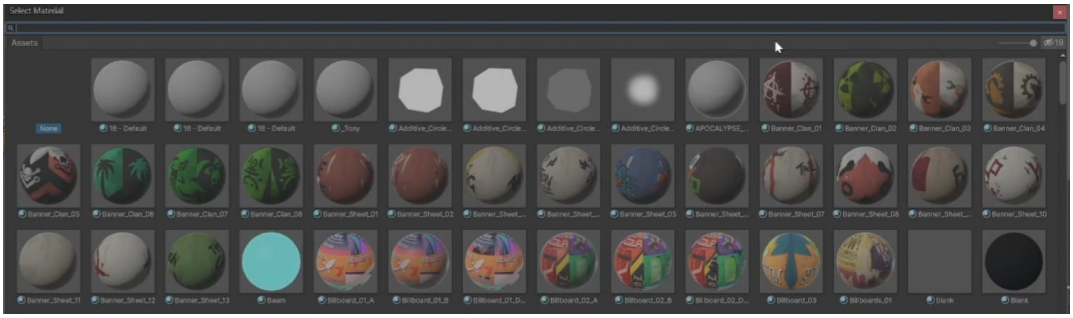
    b. Adjust **Handle Draw Distance** & **Vertex Handle Size** if the vertex handles are too cluttered or too sparse.

    c. Under *Gap Groups*:
        i. Add a new group (Add `Group` button).
        ii. Choose `Fan` or `Convex` fill mode.
        iii. (Convex only) Tweak `Convex Steps`, `Convex Exponent`, and `Apex Offset`.

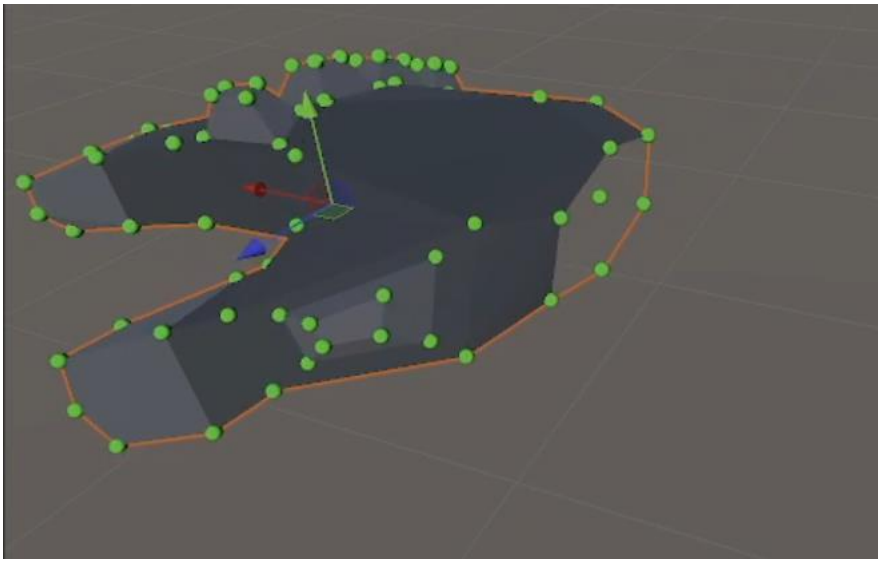
iv. (Optional) **Use Secondary Material** if you want a separate material on the filled geometry and choose the flip direction of the material.

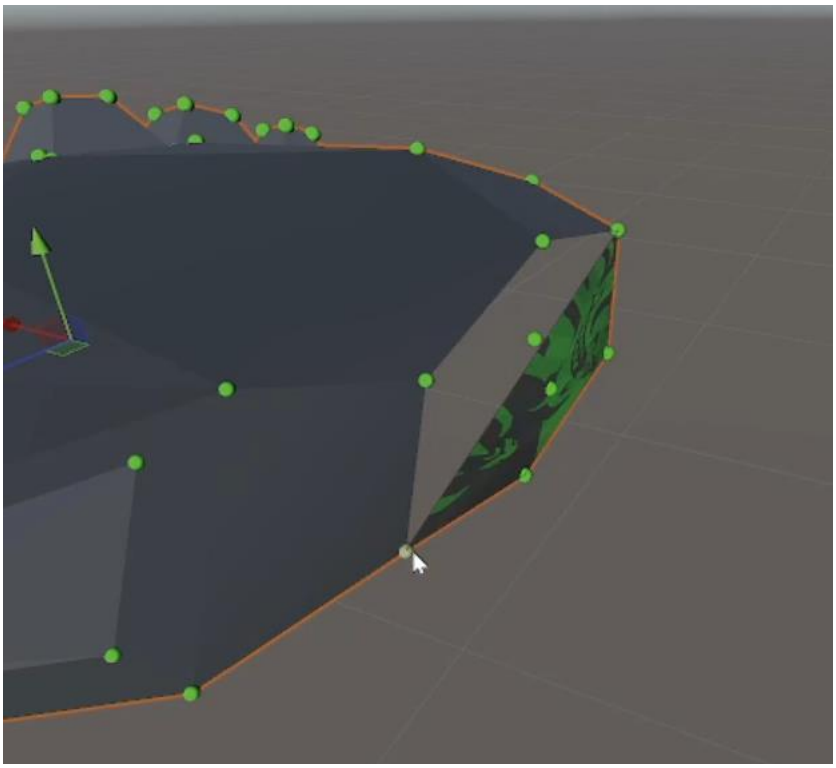


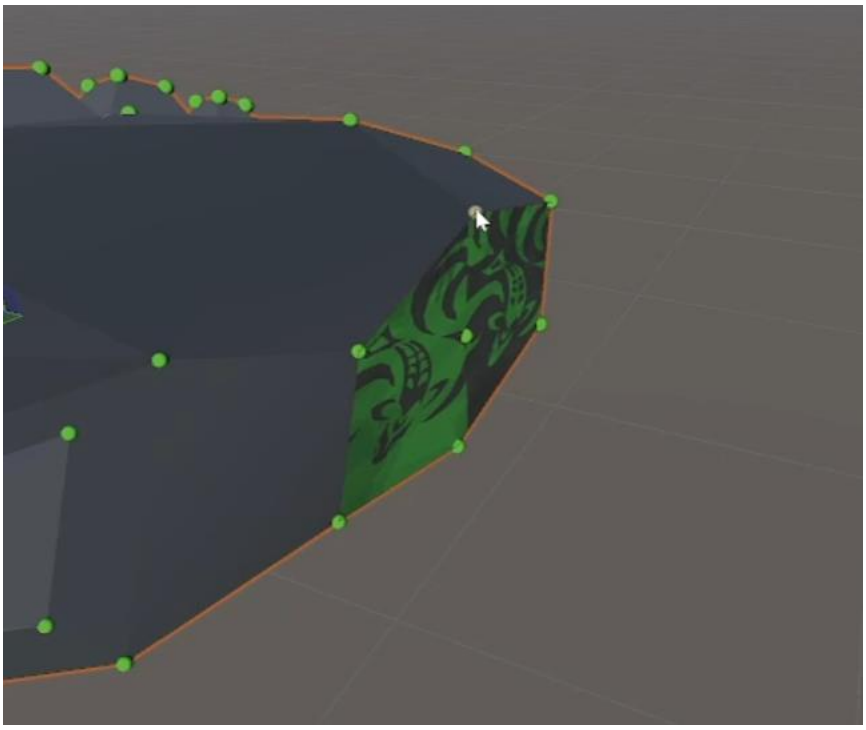




d. **Mark Boundary Vertices** in the SceneView:
   i. Click a group's foldout to expand (making it the "active group").
   ii. Rotate/zoom your SceneView to see the hole's boundary vertices.
   iii. Click each relevant vertex handle to assign it to the active group's boundary.
e. Optionally, add more groups for multiple holes.







4. **Update Preview** whenever you want to see how the fill geometry looks.
5. **Finalize** when you are satisfied:
    a. A new `SkinnedMeshRenderer` is created, named `[OriginalName]_Filled`.
    b. The result is 'saved' on the prefab/gameobject it came from. **Note**: If the selected gameobject with the skinned mesh renderer component came from a prefab, override the prefab so that the skinned mesh is saved onto the prefab gameobject. If it's not a prefab, you should now make the gameobject a prefab.
    c. A static mesh is also created and placed under your chosen folder **/StaticMeshes**

7. Skinned Mesh Combiner

7.1 Overview

One of the major issues when having multiple skinned meshes in a scene is the performance hit. With the **skinned mesh combiner** tool you can easily recombine your separated skinned meshes back into a single skinned mesh.

The **Mesh Rebaker Tool**:

- Takes in a single prefab from the scene with skinned meshes and combines them outputting a single skinned mesh.

7.2 Interface and Settings

Open via **Tools → MeshForge → One Shots → Skinned Mesh Combiner**.

Below are the primary fields and options in the **Skinned Mesh Combiner Window**:

1. **Save Folder** (Required)
    1. A text field labeled "Save Folder (Required):"
    2. Sets the subfolder path under `Assets/` where the output mesh will be saved.
    3. Example: If you enter `MyGame/Meshes`, the final path becomes `Assets/MyGame/Meshes`.
2. **Source Prefab** (Required)
    a. A field that takes in a prefab that is placed in the scene. This will be the prefab that is searched for skinned mesh renderers to combine, it will also be used to place the combined skinned mesh
3. **Override Prefab After Combine?**(Optional/Toggle)
    a. If marked true, the prefab the combined mesh is placed on will automatically be overridden.
4. **Combine Skinned Meshes**(button)
    a. Triggers the combining operation, adds the combined mesh to the source prefab

7.3 Skinned Mesh Combiner Quick Start

1. **Open Tools → MeshForge → One Shots → Skinned Mesh Combiner.**
2. **Drag prefab with skinned meshes into scene.**
3. **Drag prefab from scene to Source Prefab field**
4. **Press Combine Skinned Meshes**

8. Topology Splitter

8.1 Overview

With the addition of ‘static mesh’ support the ‘Topology Splitter’ aims to provide a more robust editing tool for nuanced decisions when splitting a static mesh built on a ‘Topology Separation’ algorithm that separates the mesh into individual parts.

The **Topology Splitter Tool**:

- Takes in a single game object from the scene containing a MeshFilter/MeshRenderer combo and splits the mesh into individual parts. The user is then given options for how they want to merge, group, and cap each individual part.

8.2 Interface and Settings

Open via **Tools → MeshForge → Topology Splitter**

Below are the primary fields and options in the **Topology Splitter Window**:

1. **Save Folder** (Required)
    a. A text field labeled “Save Folder (Required):”
    b. Sets the subfolder path under `Assets/` where the output mesh will be saved.
    c. Example: If you enter `MyGame/Meshes`, the final path becomes `Assets/MyGame/Meshes`.
2. **Source Object** (Required)
    a. A field that takes in a game object that is placed in the scene or from the assets folder. This object musst contain a combination of a Mesh Filter and Mesh Renderer to be used with the Topology Splitter.

3. **Settings Dropdown**(foldout)
    a. **Overwrite saved objects**(toggle): Used to overwrite an object in the asset folder with the newly created object. Uses name comparison.

If turned off this will create an entirely new object and save it to the designated folder.

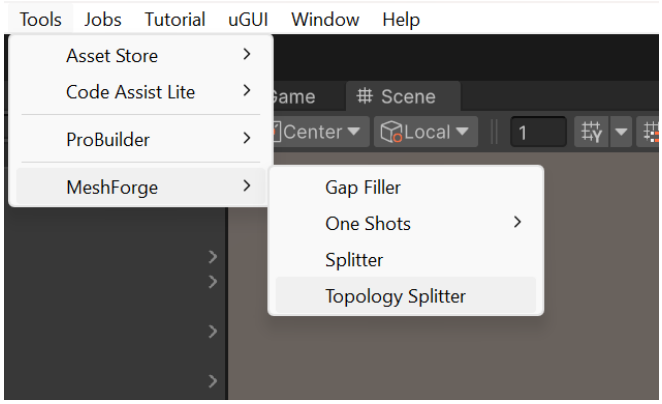
b. **Allow Ungrouped parts to be used during finalization**(toggle): If on the ungrouped parts will follow all the rules set by the settings/user and will be in the final generated game object. If off all ungrouped parts will not be added to the final game object and will not be able to be prefabbed.
c. **Save Meshes**(toggle): If on this will allow for the meshes of all prefabbed parts to be saved. If it is off this means the mesh for each part will not be saved. This means that all options for prefabbing will be unavailable, and only the scene instanced object can be used.
d. **Prefab main container**(toggle): Used for prefabbing the main container that holds all the parts. It is recommended this is only turned off if you plan to 'prefab all individual parts' or save specific parts (see Prefab Group Parts in Group Settings).
e. **Prefab all individual parts**(toggle): Used for prefabbing every single individual part that is separated from the original mesh. It is off by default because having it on may generate unwanted 'junk' prefabs i.e. creating prefabs for all individual screws from a machine.
f. **Cap separated objects**(toggle): Applies a cap to each gap found in the individual parts meshes. It is recommended that this stays on unless the user wants to create a custom cap for the mesh using the '**Gap Filler**'
g. **Add physics components**(toggle): If on it adds physics components (rigidbody /mesh collider) to each individual part. Mesh Collider is used as it is the most accurate when using colliders for separated parts.
h. **Draw ungrouped parts**(toggle): If on all of the triangles that belong to ungrouped parts will be drawn.
i. **Ungrouped color**(color selector): Use this to set the color that will be used for all ungrouped parts triangles.
j. **Triangle draw distance**(slider): Sets the distance at which ungrouped parts triangles will be drawn to the scene view.
k. **Triangle handle z offset**(slider): Sets the distance away from the meshes surface in the forward direction the triangle will be drawn. The higher the value the further away from the surface of the mesh each triangle will be drawn.

4. **Split Preview Mesh**(button)

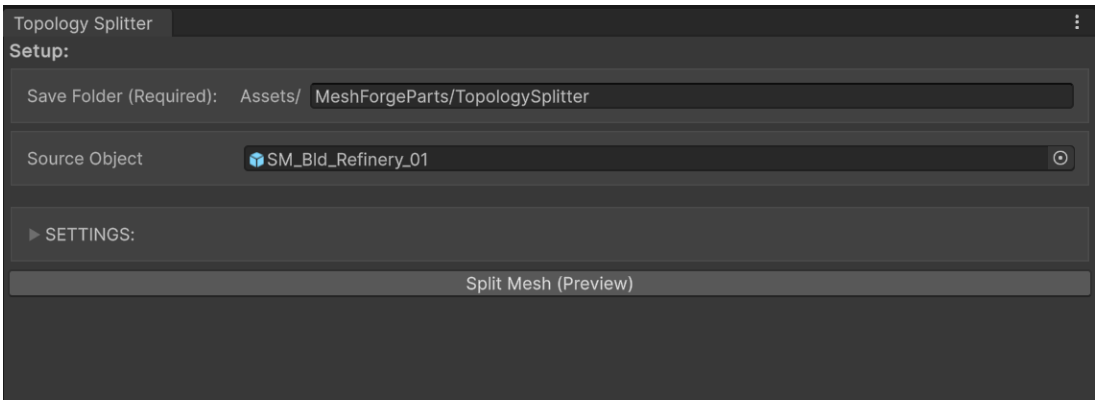
    a. Takes the source object and creates an instance of it in the scene and splits it using the 'Topology Separation' algorithm. The rest of the topology window is refreshed to display the new settings.
5. **Ungrouped Preview Parts**(foldout)
    a. Displays all of the ungrouped parts created from the source object.
6. **Groups**(foldout)
    a. Displays all the groups that the parts are placed into
    b. **Group**(nested foldout): Displays specific groups parts and a dropdown for Group Settings.
    c. **Settings**(nested foldout): Contains all of the settings for the specific group.
        i. **Group Color**(color selector): Sets the color that will be used when drawing the groups specific triangles.
        ii. **Secondary Material**(material field): If one is added, the group will use that material as the cap material when capping the parts meshes gaps.
        iii. **Merge Group**(toggle): If turned to true the parts in this group will be 'fused' creating a single mesh with those specific parts meshes.
        iv. **Draw Triangles**(toggle): If turned on, the triangles for this groups parts will be drawn to the scene view. If turned off, this groups parts triangles will not be drawn.
    d. **Add selected to this group**(button): If a part or parts are selected in the scene hierarchy and this button is pressed then the parts will be added to this group.
7. **Add Selected to New Group**(button)
    a. If a part or parts are selected in the scene and this button is pressed, then they will be added to a newly created group.
8. **Remove Selected From Current Group**(button)
    a. If a part or parts from a single or many groups are selected and this button is pressed, then they will be removed from their current groups.
9. **Finalize Meshes**(button)
    a. If pressed, generates an entirely new game object to the scene with the requested settings applied and parts organized by group if applicable.
10. **Clear Preview**(button)
    a. If pressed, clears the entire instanced preview and clears all groups.

8.3 Topology Splitter Quick Start

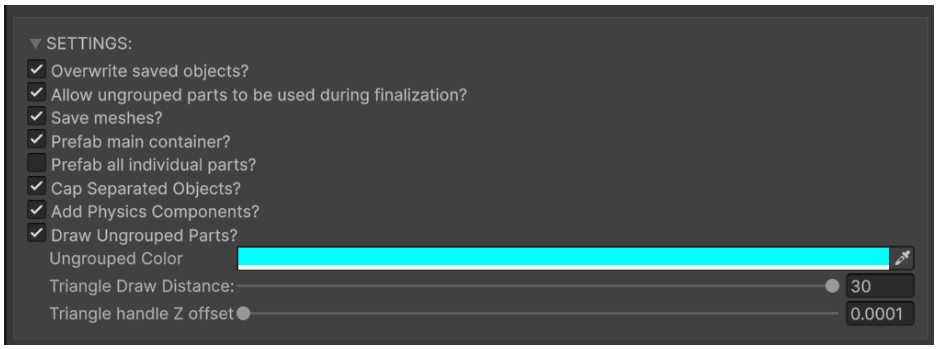
1. **Open Tools → MeshForge → Topology Splitter.**



2. **In the Topology Splitter window add a object into the source object field**
   a. Drag and drop a game object from the scene or from the assets folder. The game object must have a mesh renderer and mesh filter to be valid.

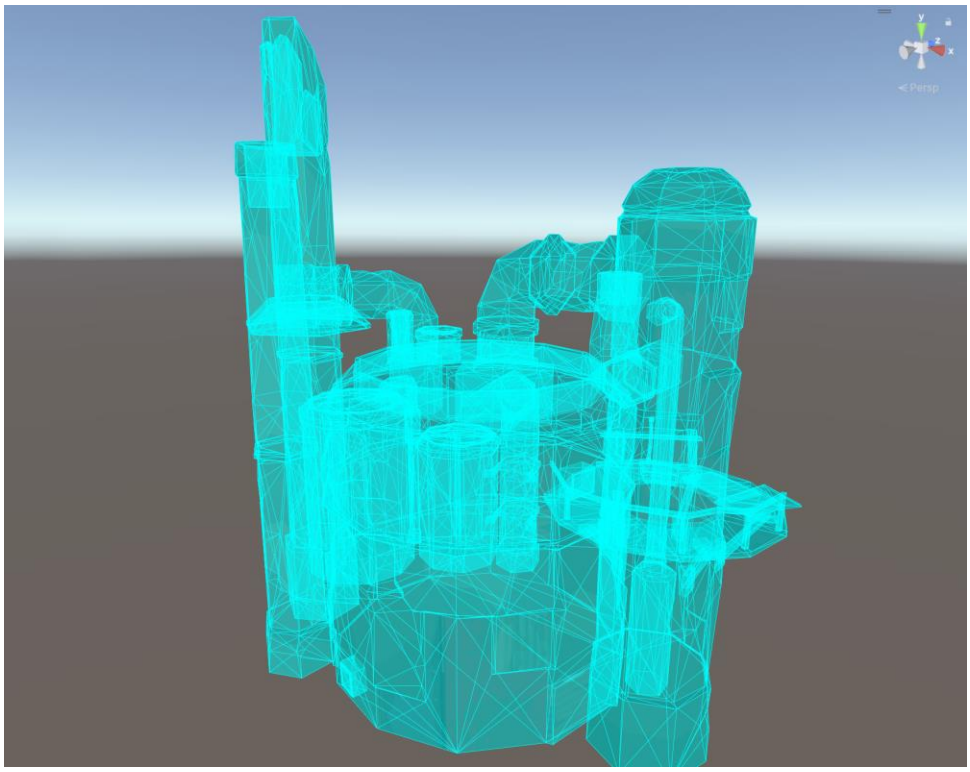


3. **In the settings foldout ensure the settings are set appropriately for this simple demonstration run (defaults).**

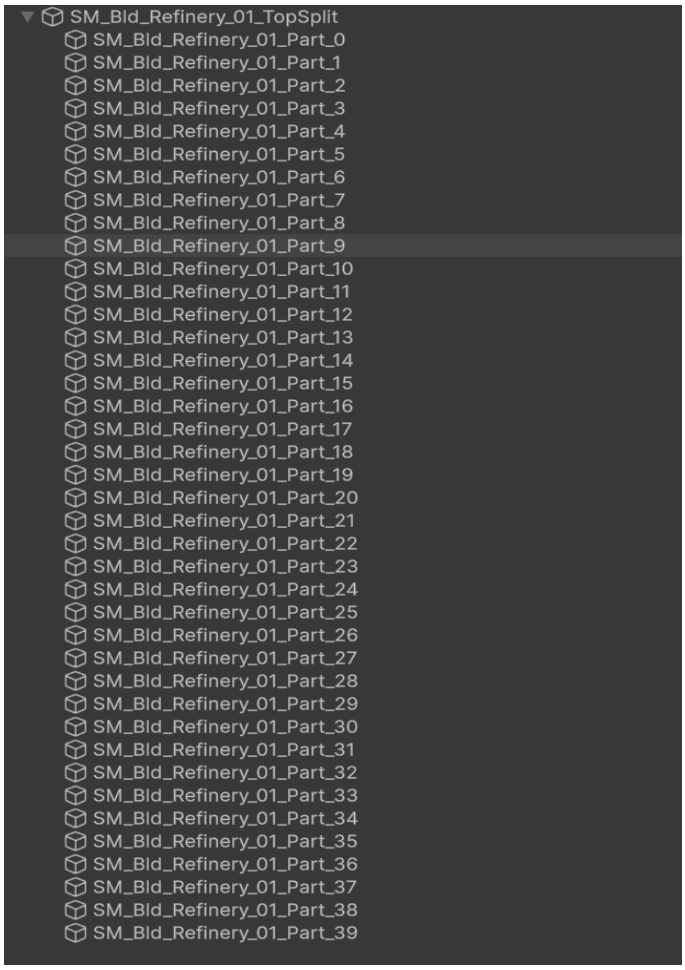


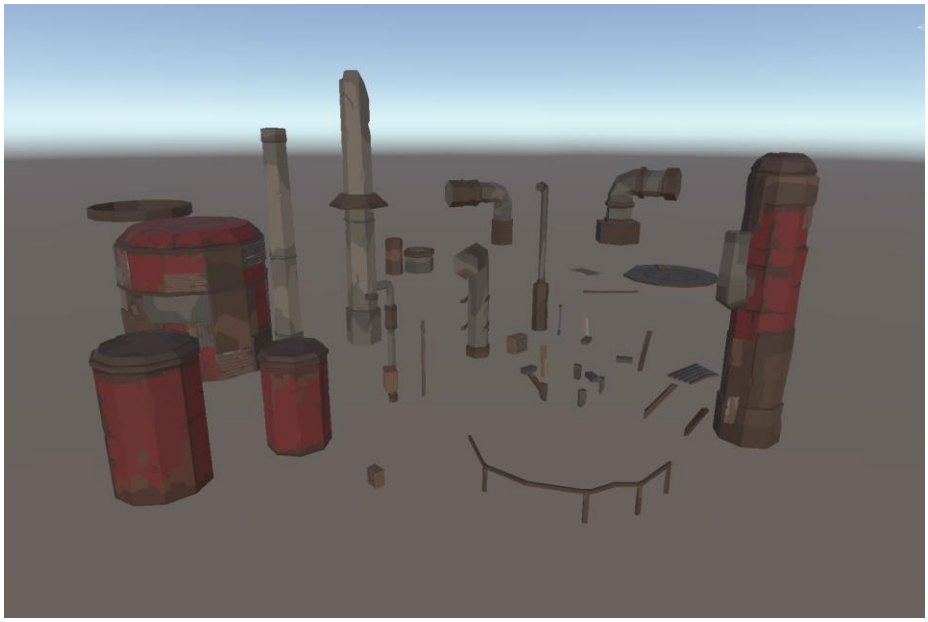
4. **Press Split Mesh (preview) button**

a. This will take your game object and instantiate a preview game object in the scene with all the parts that were separated.



5. **Press Finalize Mesh button**
    a. You should now notice that the example object was replaced with an object that represents the final product.
    b. The prefabbed final product and saved meshes can be found in the designated file locations.


SM_Bld_Refinery_01_TopSplit
SM_Bld_Refinery_01_Part_0
SM_Bld_Refinery_01_Part_1
SM_Bld_Refinery_01_Part_2
SM_Bld_Refinery_01_Part_3
SM_Bld_Refinery_01_Part_4
SM_Bld_Refinery_01_Part_5
SM_Bld_Refinery_01_Part_6
SM_Bld_Refinery_01_Part_7
SM_Bld_Refinery_01_Part_8
SM_Bld_Refinery_01_Part_9
SM_Bld_Refinery_01_Part_10
SM_Bld_Refinery_01_Part_11
SM_Bld_Refinery_01_Part_12
SM_Bld_Refinery_01_Part_13
SM_Bld_Refinery_01_Part_14
SM_Bld_Refinery_01_Part_15
SM_Bld_Refinery_01_Part_16
SM_Bld_Refinery_01_Part_17
SM_Bld_Refinery_01_Part_18
SM_Bld_Refinery_01_Part_19
SM_Bld_Refinery_01_Part_20
SM_Bld_Refinery_01_Part_21
SM_Bld_Refinery_01_Part_22
SM_Bld_Refinery_01_Part_23
SM_Bld_Refinery_01_Part_24
SM_Bld_Refinery_01_Part_25
SM_Bld_Refinery_01_Part_26
SM_Bld_Refinery_01_Part_27
SM_Bld_Refinery_01_Part_28
SM_Bld_Refinery_01_Part_29
SM_Bld_Refinery_01_Part_30
SM_Bld_Refinery_01_Part_31
SM_Bld_Refinery_01_Part_32
SM_Bld_Refinery_01_Part_33
SM_Bld_Refinery_01_Part_34
SM_Bld_Refinery_01_Part_35
SM_Bld_Refinery_01_Part_36
SM_Bld_Refinery_01_Part_37
SM_Bld_Refinery_01_Part_38
SM_Bld_Refinery_01_Part_39




Assets > MeshForgeParts > TopologySplitter > SplitStaticPrefabs
SM_Bld_Refiner...


Assets > MeshForgeParts > TopologySplitter > SplitStaticMeshes
SM_Bld_Refiner...

9. Best Practices & Tips

- **When using assets try to only modularize meshes across assets made by the same 3D artist/ company**: They will typically maintain the same naming conventions/ bone hierarchies that allow for easier modularization.
- **Use the correct direction when selecting vertices during gap fill:** If selecting vertices with the scene view camera **outside** of the mesh, select vertices in a **clockwise** order. If the scene view camera is **inside** the mesh, select in a **counterclockwise** order.
- **Backup Often**: Always keep a backup of your original mesh assets.
- **Want to remove parts of a Mesh? (Accessories, etc...):** If you only want to remove portions of the mesh like a set of glasses or hair **where using the fill algorithm would be tricky.** Use paint mode, paint mode allows you to group triangles and have those triangles finalized without the need for using the fill algorithms.
- **Missed Triangles:** If you accidentally miss triangles on a part of your skinned mesh when splitting, send the split mesh back through the splitter add a new group called “Garbage” use the paint tool to select any triangles that you want to remove. And when finalizing make sure to use “allow ungrouped meshes”. You will end up with a mesh that doesn't have floating triangles!
- **Refine, Refine, Refine!:** There is no limit to how many times you can run a mesh through the gap filler, or the splitter use both tools in creative ways to refine your skinned mesh to be what you want it to be.

10. FAQ/TroubleShooting

1. **Q**: *Why are some vertices / triangles not visible?*

A: Check your Handle Draw Distance—the vertex or triangle might be outside that range.

2. **Q**: *Why is my new static mesh missing textures when gap filling?*

A: When selecting the skinned mesh renderer object in the scene, if that skinned mesh renderer is using materials that are 'instances', that instance will be unable to be added to the static mesh for saving. You will need to manually reapply the material to the static mesh.

3. **Q**: *Why is it when i am using the splitter and i press separate and save none of my groups are in the final prefab?*

A: One possible reason is that you used the split mesh tool and separated parts of the mesh using the wrong mode for the group. If you do not use the fill algorithm to "process" a group then you have to use paint mode. Paint mode triangles allow for the user to freely select triangles and add them to the group without the use of the fill algorithm. Make sure you are in the right mode when selecting triangles. Boundary mode is for creating boundaries, seed mode is for creating seeds, paint mode is to freely add triangles to a group.

4. **Q**: *Why can I not 'drag select' triangles?*

A: This has to do with the unity version you are using. Due to event propagation and consumption in the editor in older versions of unity the drag select will not work as intended. Update the unity version to take full advantage of drag select. Drag select has been tested in 2022+. Another more intrusive option is in the Splitter Window script OnSceneGUI method place this after the two 'if(conditional checks)' before the 'Event e = Event.current;' statement:

```
// Get a unique control ID.
int controlID = GUIUtility.GetControlID(FocusType.Passive);
// In the Layout event, register this control to capture events.
if (Event.current.type == EventType.Layout)
{
    HandleUtility.AddDefaultControl(controlID);
}
```

be aware that by placing this in the script Unity's built in tools i.e. the mover tool, will not be able to process events until the splitter window is closed.

5. **Q**: *Can I use non skinned meshes on the Gap Filler or Splitter tools?*

A: Yes, switch to the appropriate tab in either tool to split either skinned or static meshes.

6. **Q**: *Why is it when I move my preview object it sometimes gets 'stuck' becomes immovable?*

A: This is due to the handles and the move tool fighting for control during OnSceneGUI. One quick fix is to move the scene view camera outside of the handle draw distance range then move the preview object to your ideal location. Last move the scene camera close to the preview object to begin working on it again.

5. **Q:** Will this tool work with quadrupeds?

A: While not explicitly marketed for more than humanoid characters I have been able to successfully modularize some quadruped's assets. The issue comes down to the complexity of the bone hierarchy and what the target bone hierarchy looks like. Trying to modularize vastly different hierarchies to one another becomes much more difficult to put into an easy-to-use workflow.

6. **Q:** Will this tool work with Hi-Polygon Characters?

A: Yes, most of the demonstrations are on low poly characters because it is the easiest to quickly demonstrate the functionality of the tools as there are less triangles/ vertices that need to be worked with/selected, and the mesh complexity is much lower. **However, note that your performance may vary. To avoid pipeline dependencies, I avoided the use of shaders and instead decided to use unity's built in handles library to show the selection of triangles/ vertices. To help with performance I implemented a z-axis view threshold and a range for drawing the handles to the scene view.**

11. Version Updates

MeshForge Update v1.2.0 (The Static Mesh Update):

This update includes major changes to the 'Splitter' tool, 'Gap Filler' tool, and adds a new editor window called 'Topology Splitter'. Both the 'Gap Filler' and 'Splitter' tools now have static mesh capabilities. The new 'Topology Splitter' tool allows the user to split a static mesh based on topology allowing for what is essentially a "breaking" of all the objects that make up a mesh.

File Updates:

- Updated default save file directories to be specific to the window the object being saved is being created from, e.g. Splitter, Gap Filler, etc.…
- Moved Combiner and MeshRebaker windows under a new menu item path called 'One Shots' to emphasize their use case and to better organize for future one shots that are planned.

Bug Fixes:

- Fixed bug where 'resetting groups' on the Splitter would not remove the groups from memory.

Changes to UI:

Splitter UI-

- Added static Mesh Tab includes all static mesh related options/ functionality.
- Added setting for triangle Z offset.
- Moved all settings to be placed inside the group management block.
- Added foldouts to nest groups and nest settings into.

Gap Filler UI-

- Added static Mesh Tab includes all static mesh related options/ functionality.
- Moved all settings to the top of the 'Gap Filler' window.
- Added foldout for settings.
- Added option for including physics components on the static prefabs.

Quality of Life/Performance:

- Improved Handles performance on the splitter and Gap Filler allowing the user to display more handles (triangles/vertices) at one time with a lower performance cost.
- Updated job functionality/chaining in preparation for runtime capabilities.

MeshForge Update v1.1.0 (The Automation Update):

This update includes a new group creation algorithm for the splitter allowing users to set a configuration for bone weights and automatically create groups based on bone weights. Automation was brought to the gap filler. Now, when filling gaps, the user can select multiple skinned meshes at once and simply select the automation settings they want and press the create groups button to create/ fill groups for all gaps in all meshes. The user will still have control to go through and add changes even after the groups are made though this is limited to material updates and adding new groups/vertices selections.

File Updates:

- Added new scriptable object type called 'BoneWeightGroupConfig', includes options for adding group mappings e.g. GroupName (the name set on the group in the splitter), Common keywords(Words used to find specific bones for the group),Left keywords (Words used to find specific bones for the group that belong to the left side),RightKeywords (Words used to find specific bones for the group that belong to the right side).

Changes to UI:

Splitter UI-

- Included a field for the 'BoneWeightGroupConfig' file.
- Added ambiguity threshold slider for setting smooth boundaries between bone weight groups.
- Added Advanced Grouping button for automatically creating groups using bone weights.

Gap Filler UI-

- Added toggle for group automation.
- When group automation toggle is selected all the options for the group automation appear, similar to the gap group settings visually.
- Added button called 'create groups' when clicked all the groups of the selected skinned meshes will be filled with the selected automation settings.
- Added color-based dividers to differentiate between the selected objects list of groups when multiple objects are selected.

Bug Fixes:

- Fixed bug in gap filler where when removing a group an error would appear for GUI Layout.

Quality of Life:

- Made it so that the user no longer has to add a group when splitting a mesh. making it so the user can quickly do things like deformation without the step of adding a group.

12. Conclusion

Thank you for using **MeshForge**. With its **Mesh Rebaker**, **Splitter, Gap Filler, Skinned Mesh Combiner, and new Topology Splitter** tools, you can quickly generate modular mesh pieces without losing their original state and addressing tricky geometry holes. Assets used in the images above are for educational purposes and are not included in the asset.
For additional help or feature requests, please reach out to the developer via email: **anthony.halstead@carryforgestudio.com**