

Building a Character

How to use the Linteum character assembler to create a character

Preface

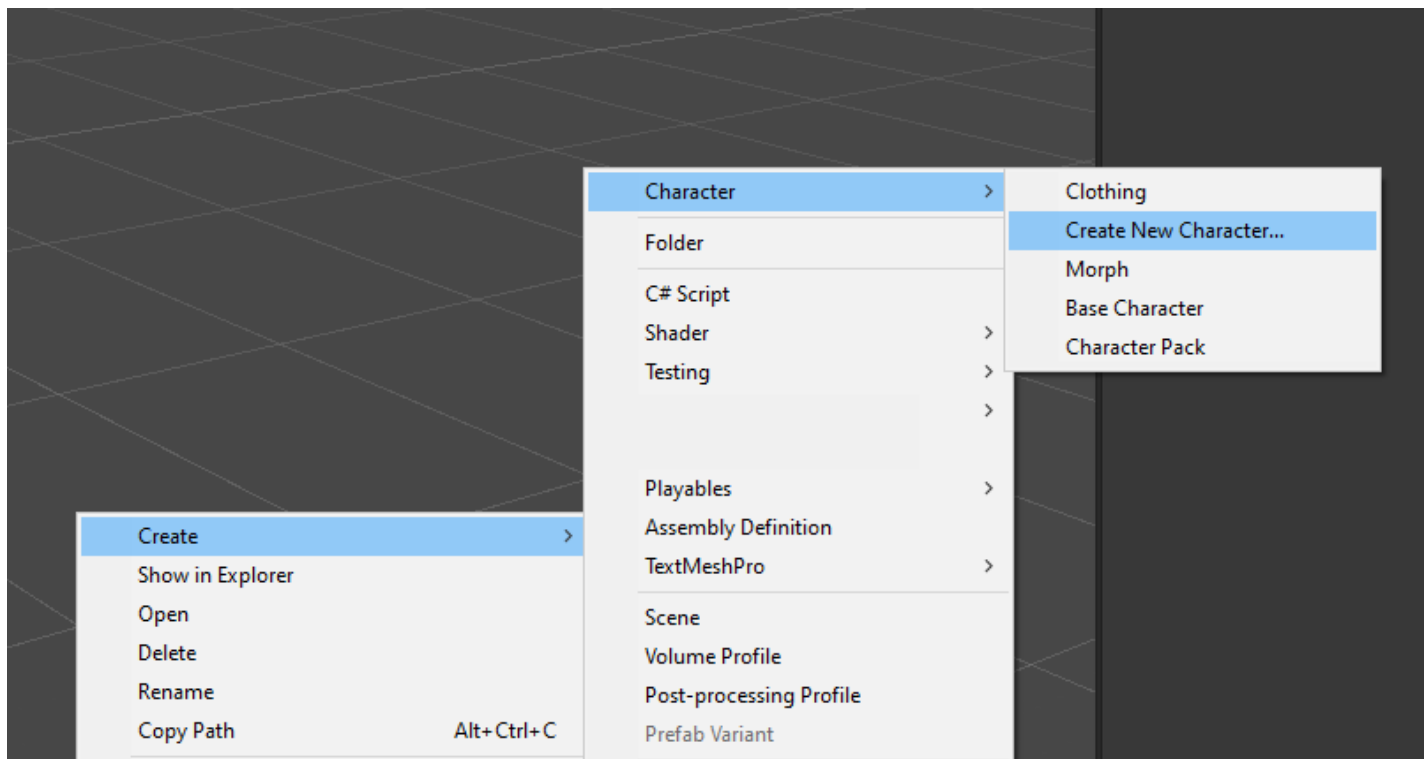
Characters, clothing and other items compatible with this system are intended to be used through the Character Kit somewhat exclusively. While source assets have been provided for advanced users, directly editing the "source" files of assets is not recommended. An example of this, the included Materials are configured for HDRP - when you assemble the character (see below), these will be copied, and the new copies will be rewritten for the desired rendering pipeline.

The Character Kit is designed to be both easy to use, and powerful - and takes care of a number of complex problems, such as merging bones and skinned mesh renders onto a common skeleton, copying morphs into clothing and removing unseen surfaces from the character.

Building a Character

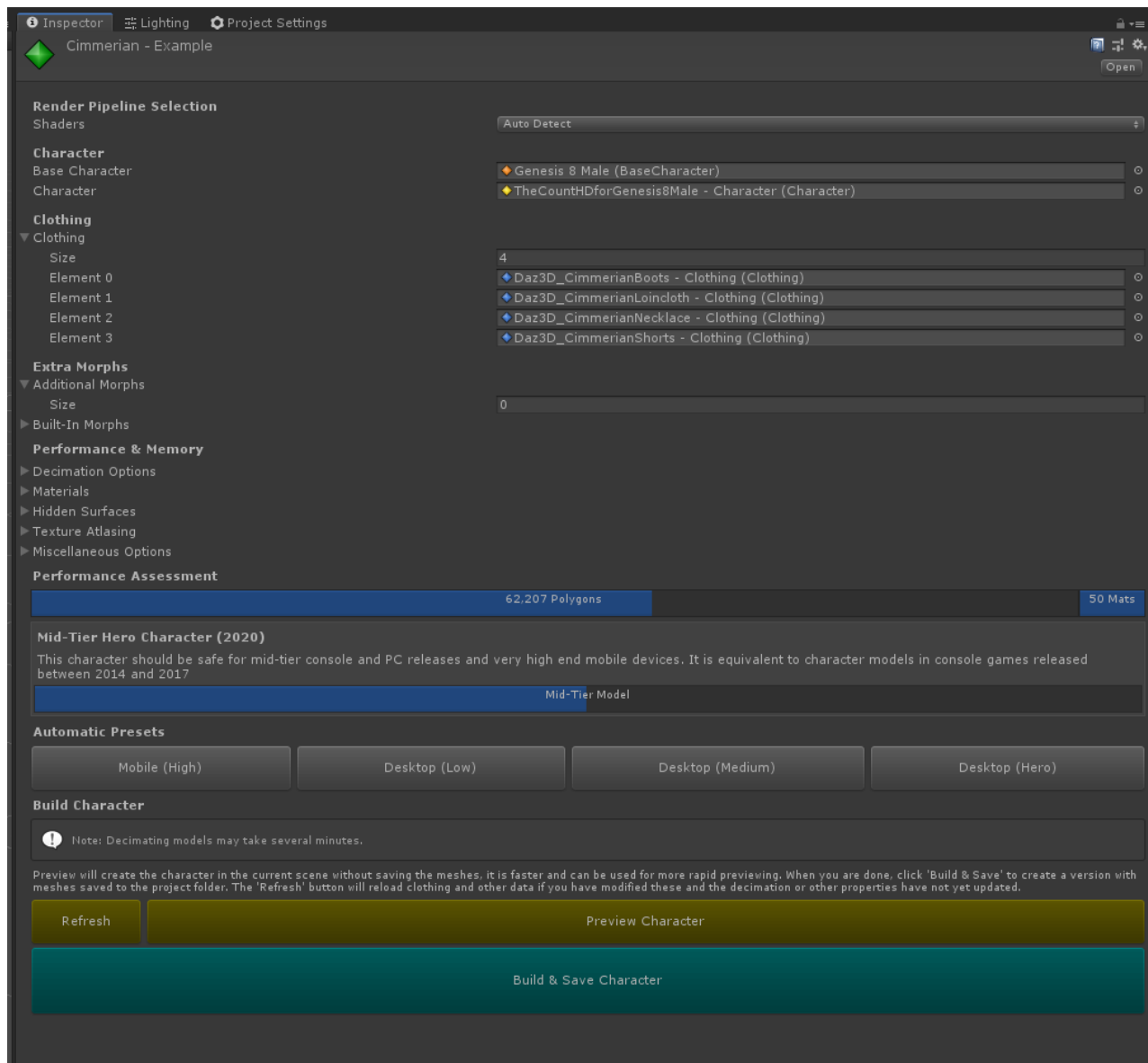
To build a character right-click in your Project window and select

Create -> Character -> Create New Character... this will create a new "Character Assembler" asset, which stores the settings of your new character, so you can duplicate and tweak your results easily.




The Character Assembler

Upon creating a new Character Assembler, your Inspector (if not visible, use `Window -> Inspector` from the Unity Main Menu) will show a selection of options to modify your character, as well as some basic performance heuristics about the assembled result.



The Character Assembler Inspector

To start, use the  icon to the right of the 'Base Character' field, to show the list of base characters in your project. This is the 'base model' you will be assembling against - this is typically a basic humanoid. Once you have added this basic model, you can add optionally one 'Character' field, and multiple 'Morphs' and 'Clothing'



Tip: You can drag assets from the Project view onto the 'Clothing' and 'Additional Morph' fields to quickly add them to the list of equipped clothing and morphs.

Render Pipelines

Linteum will automatically detect if you have a particular render pipeline running under its default setting. Provided shaders are compatible with URP 7 (2019), HDRP 7.2/7.3 (2019) and 4.0 (2018) as well as the legacy Standard Render Pipeline. Using the 'Shaders' drop down, you can force it to one of these options.

Characters

Characters provide a set of override materials, and one base morph which allow you to change the base character in significant ways. Characters provide a quick and easy way of differentiating your models, while still retaining compatibility with the larger clothing and morph libraries.

A Note on Skeletons

Characters may also bring across broad changes to the bones and skeleton of the base character (modifying the underlying rig); while morphs in general can be modified through the use of the resulting Blend Shapes to transition to and from a character, these skeletal changes are more 'permanent', and transitioning between characters with more extreme size differences is not recommended.



Scripters: the data on these skeleton morphs is available in the BoneScales, BonePositions, BoneRotations and BoneNames array fields of the SkeletonMorph class. They represent the final local positions of the adjusted characters.

Clothing

Clothing is additional prefabs that will be attached to your character; if these prefabs are rigged against the same skeleton as the underlying base character, the bones and rigging will be transferred to use the original base skeleton, ensuring that the clothing moves with the character itself.

Extra Morphs

Extra Morphs provide the ability to create more extreme or subtle variations of a character and their expressions. It is worth noting that extra morphs will also be copied to the final assembled clothing;

so for example, a body weight morph will ensure that the clothing resizes in accordance with the original character.

Morphs are applied as 'Blend Shapes' which are visible on all Skinned Mesh Renderer components in the finished result. You can dial them up or down at runtime through modification of these blend shapes. Extra Morphs can also include things such as facial expressions, or a full FACS rig.

Note on memory use: morphs will add $1-3 \times$ the total vertex count \times 12 bytes per morph added to the character. For a 60,000 vertex character, this is around 703kB to 2.1mB of RAM use per morph; adding hundreds of morphs to a character may pose a memory problem.

Instances of the same character will not count multiple times.

Built-In Morphs

Some characters may include additional built-in blend shapes, the 'Built-in Morphs' section allows you to select which ones you wish to preserve in the finished product. Built-in Morphs that are not selected will not be included. Note these morphs are not copied to clothing.

Option: Strip Morphs

Unchecking this option will prevent any built-in morphs coming from the base character from being stripped when assembling the character.

Option: Strip Morphs From Clothing

If a clothing item includes built-in blend shapes, this option will erase them before assembling the finished result.

Performance And Memory Options

The Linteum Character SDK is designed to allow you to reduce characters and clothing designed for heavier applications and bring them into a performance profile more appropriate for runtime use.

Note that this can only be so effective without producing visual artifacts - destructive options such as decimation may require work to dial in the appropriate settings.

Decimation Options

Decimation removes and merges triangles progressive to reach a specific polygon target. Decimation is a destructive process, and results can be mixed. The decimator included in the Linteum Character kit is keyed to be rigging aware, it will try preserve triangles near joints and other complex rigging areas; this is to ensure that when animated, the result is still useful.



The Decimator options will only appear when a clothing item has been added to a character

To decimate a clothing item, drag the slider to indicate the target number of triangles you want in the final result - be aware that due to hidden surface deletion, the final number may be lower than this for the Base Character. A setting of '0' is treated as "Don't decimate".

Decimation works best on items which are broadly smooth, complex items such as jewelry, furs or rough elements will not decimate as well as other items. Further, decimated items may not work well with complex morphs, as the vertex density that the character may rely on, will be affected. Finally, it is not recommended to decimate the base character, as the resulting UV shifts can produce unpleasant results.

Materials - Merge Duplicate Character Materials

This option will examine materials attached to a character to see if they are identical (every property on the material will be compared for an exact match), if they are identical, then the materials will be merged into a single material.

Materials - Merge Duplicate Clothing Materials

This option will examine materials attached to a clothing item, and works substantially the same as character materials - if the clothing item materials are identical, then the materials will be merged into a single material.

Hidden Surfaces - Delete Hidden Surfaces

This option will probe the surface of the resulting mesh to test if the base character skin is occluded by clothing; if a triangle on the character is fully obscured by a clothing item, the hidden surface scan will remove it from the resulting mesh. Be aware, enabling this option is slow, and will cause the final character build to take several minutes.

This option can be useful in the case of animation, where the hidden surfaces can poke through when rigging is different between clothing and character models; or as a result of decimation or other changes.

Hidden Surfaces - Range Min and Max


These two options control how far above and below the surface the tool will examine for a clothing item, it is not recommended to change this from the default values.

Hidden Surfaces - Unwind Hidden Surfaces By One

This option will restore triangles neighbouring a solid section of the item, this can help preserve areas which are too aggressively deleted. This will however increase the total number of triangles in the final result, and preserve more of the overall skin.

Texture Atlasing - Atlas Textures

This option will examine each item for materials which utilize the same shader and associated settings. In the event that the materials are compatible, it will merge all the textures into fewer textures, and rewrite the UV coordinates of the final result to ensure that the result is more performant with fewer materials.

 **Compatibility note:** For atlasing to work, a material must be using one of the included Shaders, and the settings need to be compatible such that the item can be atlased. For some shaders, a more broad compatibility is achieved through the use of a packed LUT, which encodes more material information into a special texture; allowing more complex materials to be atlased together.

For more information on texture atlasing, and achieving the best results, see [this page](#)

[→ Texture Atlasing In Depth](#)</advanced-topics/texture-atlasing-in-depth>

Texture Atlasing - Atlas Resolution

This controls the final output resolution of the atlas, if the resolution is less than the texels required to render the textures at full resolution, then the textures will be shrunk to achieve the target resolution.

Texture Atlasing - Strip Textures

If this option is checked, then the following "Include" settings will be used. This allows you to remove specific textures from the result, such for example on performance restricted targets you may wish to remove normal or specular maps to save memory.

Miscellaneous Options - Optimise Mesh

If this option is checked, on some versions of Unity, this will re-order the resulting triangles and vertices for optimal GPU performance. Uncheck this option if you need to ensure vertex ordering remains consistent.

Performance Assessment

The performance assessment has been tuned against a variety of professional video games (thankyou Polycount forums!) to give you feedback on how heavy a particular item is.

Note that the poly count will be aware of the decimation options you have selected, but will not factor in hidden surface removal. Likewise the Materials count does not factor in any merging or atlasing results; as a result of these items, your character may be lighter than indicated.

Overall performance for characters is a art as much of a science - modern GPUs can perform wonders in ways that previous generations could not. As a general rule, in 2020 polycount does not matter as much as you think it might, as most 3D applications are now fill rate / fragment constrained. The only way to truly assess your output is to test the results.

Automatic Presets

There are four preset buttons which provide a rapid adjustment of the character polycount and other options. These are blunt instruments, and it is recommended that you use the various options to adjust and tune your results for your finished product.

Build Character

There are three buttons in this section;

Refresh Button

This button reloads data from clothing and morphs. If for some reason, the decimation or other options is not showing a clothing item you recently changed or swapped, tapping this button will reload the data.

Preview Character

This will build the character with all the specified options, but will not write the finished assets to disk - the character will instead appear in the scene. Note, if you create a prefab of the character, it will not save correctly. This option is purely for rapid previewing of various results, without waiting for the assets to be written to disk, or clogging up your assets folder with lots of tests.

Build & Save Character

This will build the character, and save the resulting assets to disk in a variety of unity internal '.asset' formats. Creating a prefab from the character added to the scene will persist in the ways you expect it to.

Advanced Topics

Editing the Results

If you want to edit the finished product inside of a professional 3D modelling application such as Maya or Blender, the Unity FBX Exporter (available through the Package Manager) can allow you to convert a finished prefab into a FBX file which can be edited. Be aware, you may need to manually restore materials for the trip back in.

