

UE5 AnimToTexture plug-in

- **Original Post:** This is not the original post. This is just a version translated into english. Please see the original post by [@marv_kurushimay\(株式会社マーベラス\)](#) linked below.
https://qiita.com/marv_kurushimay/items/8898046ed9986bbdd6b5#%E3%82%B9%E3%82%BF%E3%83%86%E3%82%A3%E3%83%83%E3%82%AF%E3%83%A1%E3%83%83%E3%82%B7%E3%83%A5

In the UE5 City sample, SkeletalMesh was converted to StaticMesh and an animation was played. The following is an overview of the process.

For an overview, see our previous article for a summary.

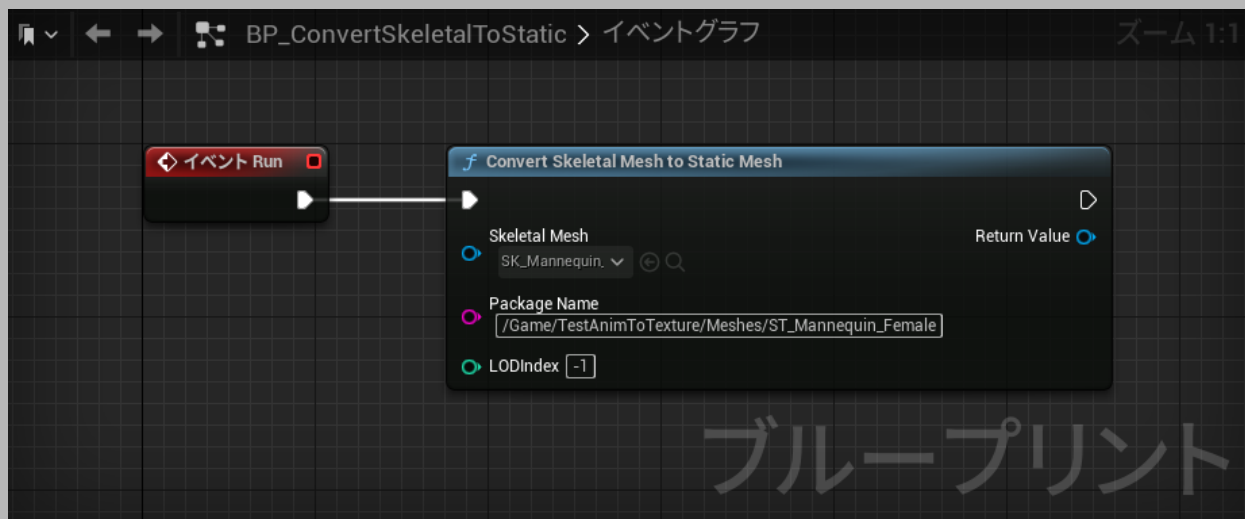
https://qiita.com/marv_kurushimay/items/b3d98f208fa494f1ddec

This article summarizes the use of the AnimToTexture plugin used in that workflow. The engine version used is UE5.0.0.

Creating a StaticMesh from a SkeletalMesh

First, create a StaticMesh for this mechanism.

It can be easily generated by calling the Blueprint function implemented in the AnimToTexture plug-in.



Create an editor utility Blueprint (*right click in content browser > Editor Utilities > Editor Utility Blueprint*), override the Run function, and place the BP function for ConvertSkeletalMeshToStaticMesh.

The arguments are

SkeletalMesh.

- Specifies the SkeletalMesh asset to convert from

PackageName

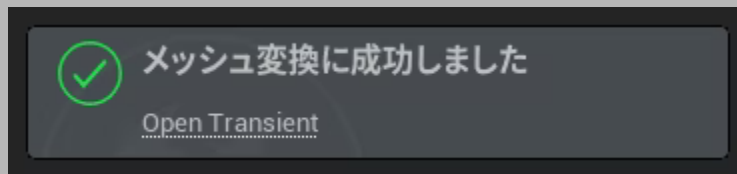
- Specifies the name of the output StaticMesh asset

LODIndex

- Specify the LOD to output to StaticMesh. If -1, all LODs that exist in the source SkeletalMesh will be output.

All that remains is to run the editor utility blueprint.

(Run the editor utility BP by right clicking on the BP and selecting 'Run Editor Utility BluePrint')



If the generation is successful, you will get a notification like the one shown above. Also, the results of the execution are output to the log, so check the log if it does not generate successfully.

Future Improvements

This time, we decided to output a specific SkeletalMesh to a specific path, so the structure is simple.

In the future, the following changes will make the tool easier to use.

- Make it based on AssetActionUtility and perform the conversion from the right-click menu of SkeletalMesh.
- Make it a function with arguments, maybe let the user choose the PackageName and LODIndex as well.
- Make it based on EditorUtilityWidget to make it easier to set various parameters.

Burn animation to texture

In this case, we will create both VertexAnimation and BoneAnimation.

In reality, you will probably end up with one or the other depending on the look and use, in which case, please ignore the one you don't need and proceed.

Asset Preparation

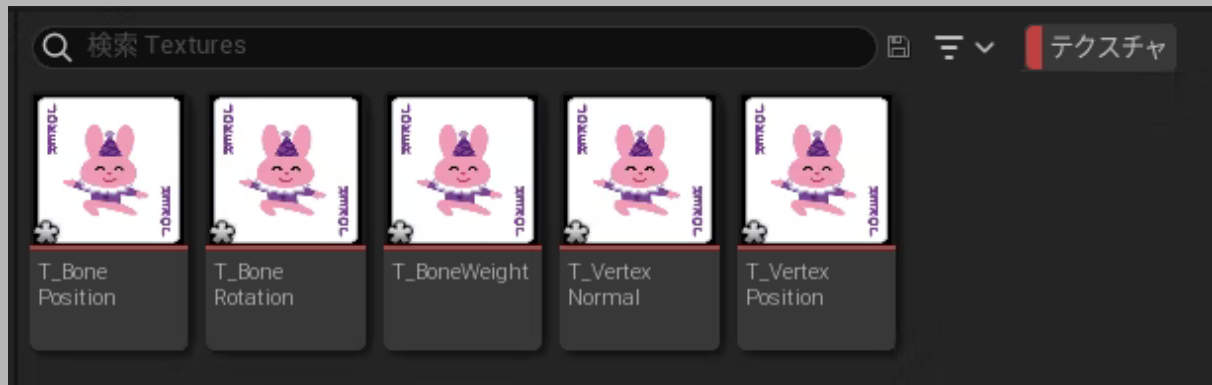
I will use the female version of Grayman (SK_Mannequin_Female) as the original asset.



(I didn't use UE5's Quinn because it was too much trouble to bring the asset to the test environment...)

Textures

First, we need to prepare the textures to which the animation will be baked: for VertexAnimation, we need two textures (Position and Normal); for BoneAnimation, we need three textures (Position, Rotation, and Weight).

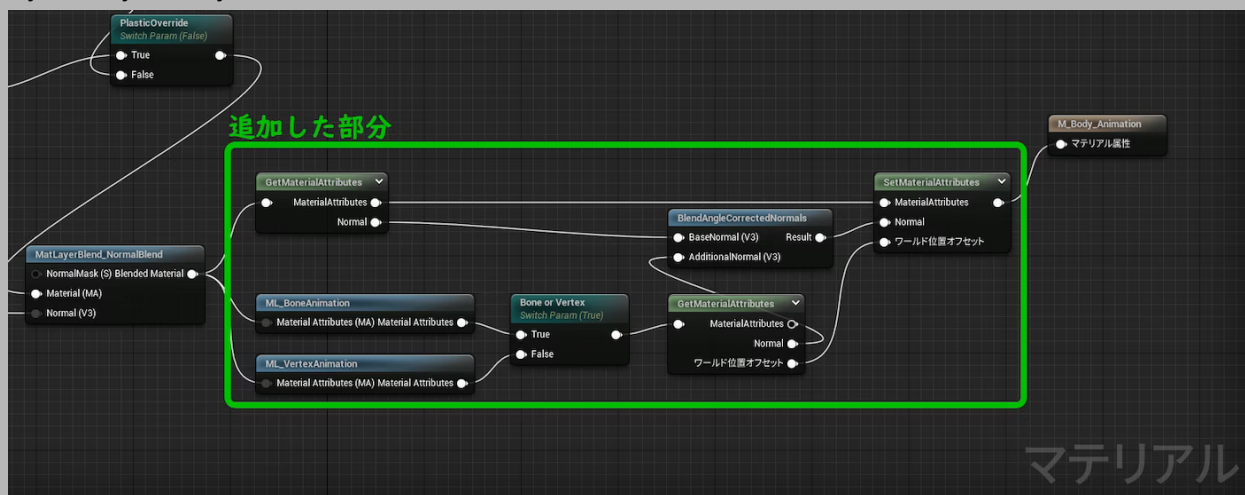


This can be just a 2D texture, but there is no way to create a texture asset in the editor that is not a RenderTarget. Therefore, import the appropriate image file to create the asset as shown above.

The size and format of the texture will be changed by itself during the conversion process to be performed later. Therefore, import an appropriate image file and create an asset.

Materials

Next, create materials and material instances that correspond to the AnimToTexture mechanism. The functions required for animation calculations are stored as material functions in the material layer, so you only need to refer to them.



Since we are using SK_Mannequin, we will use the M_Male_Body material as the base. We will add a new node, so we copied the asset and named it M_Body_Animation.

The additional part is circled in green, using /AnimToTexture/Materials/ML_VertexAnimation for VertexAnimation and /AnimToTexture/Materials/ML_BoneAnimation for BoneAnimation.

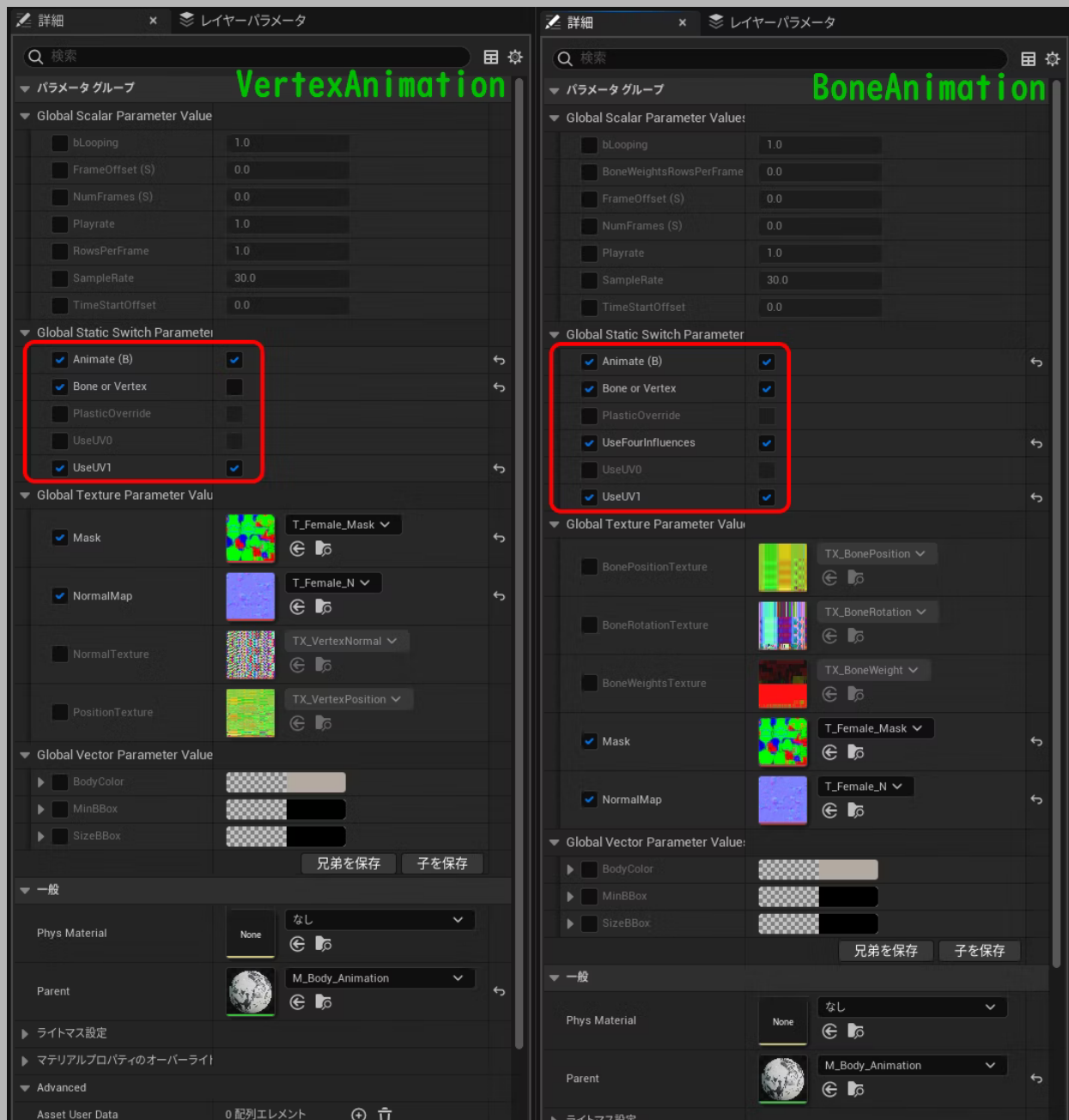
In the sample included in the AnimToTexture plug-in, VertexAnimation and BoneAnimation were created in separate materials, but this time they are branched into a single material using StaticBoolParameter.

The WorldPositionOffset output from each material function can be used as is, but the Normal output needs to be blended with the original Normal, so the BlendAngleCorrectedNormals node is used to blend them.

Material Instance

Now that the material has been prepared, the next step is to prepare the material instance. This time, we will output both VertexAnimation and BoneAnimation, so we will create four material instances by combining (Body, ChestLogo) x (Vertex, Bone).

Create a material instance with M_Body_Animation created earlier as its parent.



The areas to be modified in VertexAnimation and BoneAnimation are in the red frame.

Animate

- Set to TRUE

Bone or Vertex

- StaticBoolParameter added in M_Body_Animation
- VertexAnimation material instances are turned OFF, BoneAnimation material instances are turned ON

UseFourInfluence (BoneAnimation only)

- Same as normal BoneAnimation, but specifies the number of Bones to be influenced.
- You can choose from 1, 2, or 4, but in this case I chose 4, the same as in general BoneAnimation.

UseUV1

- Specify the UV to be referenced when calculating the animation.
- When ST_Mannequin_Female was created, the UVs needed here were added, so UV1 is used here. (UV0 is the UV for the texture)

In addition, assign an appropriate Female Mask texture and a Normal texture to the texture.

There are some parameters for animation in GlobalScalarParameterValues and GlobalTextureParameterValues, but they are not necessary here.

The necessary values are automatically set during the texture output process.

Static Mesh

Next, prepare the static mesh to which the animation will be applied.

Set the material instance you just created to the StaticMesh (ST_Mannequin_Female) created from the SkeletalMesh.

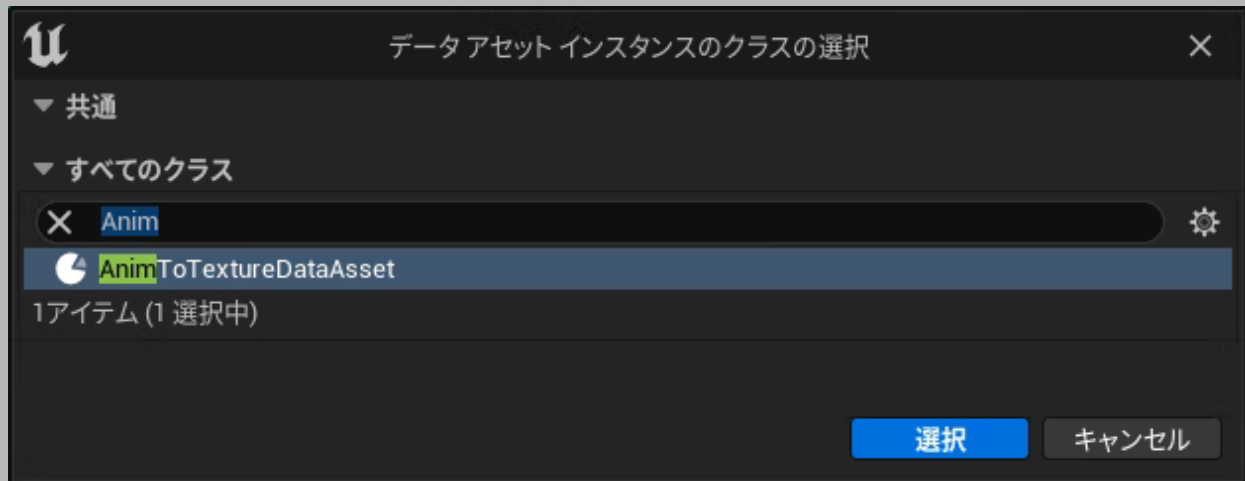
Since there are VertexAnimation and BoneAnimation, we duplicate two of them and call them ST_Mannequin_Female_VertexAnim and ST_Mannequin_Female_BoneAnim, respectively.



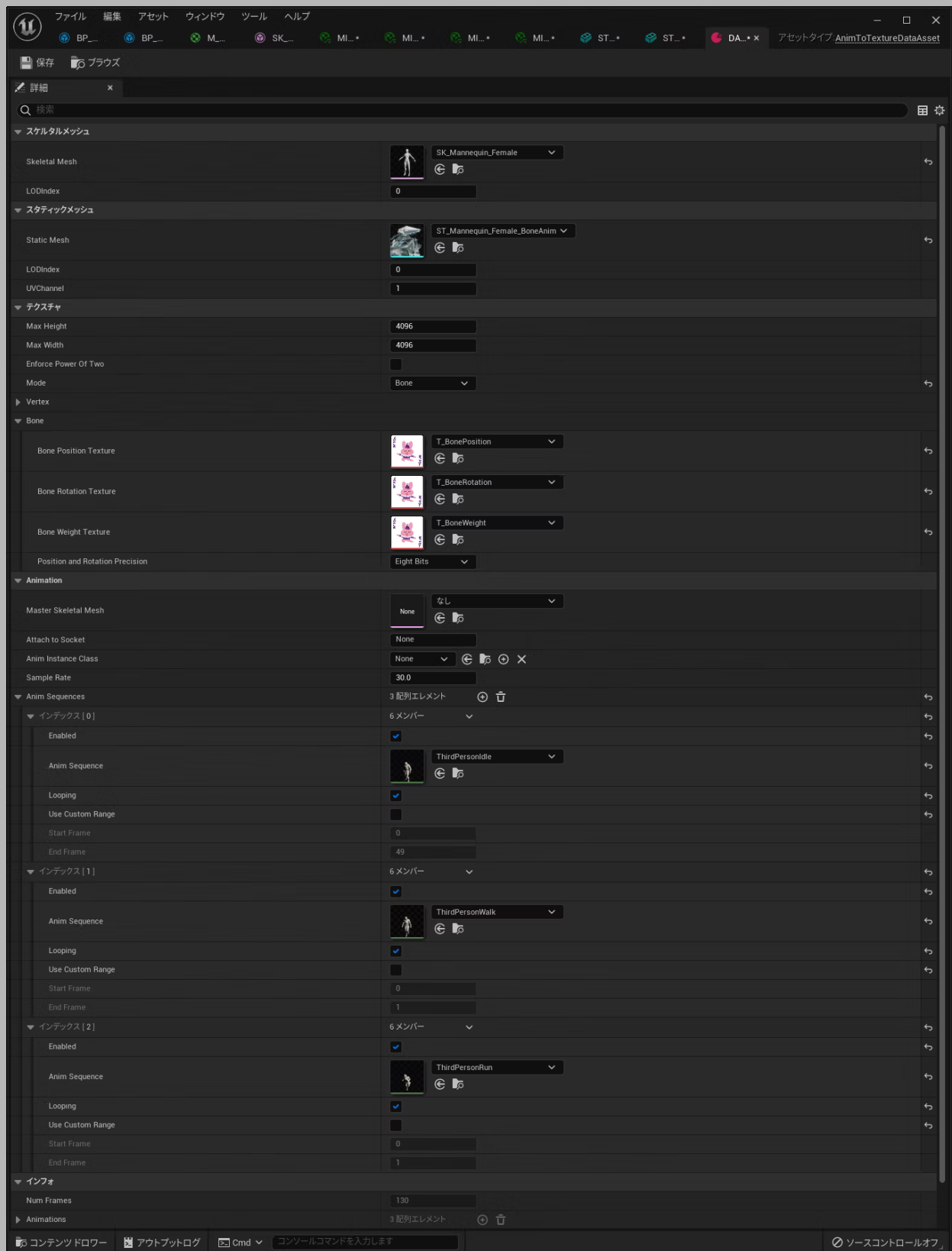
Assign a material instance to each duplicated StaticMesh. At this point, it is broken as shown above, but we do not care.

Data Assets

Specify the parameters for texture conversion from Animation in DataAsset.



Create a data asset from the content browser and specify the class as AnimToTextureDataAsset.



The above figure shows an example of BoneAnimation's data asset settings.

- Skeletal Mesh
 - Skeletal Mesh
 - Specify the original SkeletalMesh asset (in this case SK_Mannequin_Female)
- Static Mesh
 - Static Mesh
 - Specify the StaticMesh asset that you just duplicated (in this case, ST_Mannequin_Female_BoneAnim).
 - It is important to specify the StaticMesh to which the Animation-compatible material instance is assigned.
 - If you specify ST_Mannequin_Female without replacing the material, it will not be converted correctly.
- UVChannel
 - Specify 1 to use the UV1 specified in the material instance.
- Texture
 - BonePositionTexture / BoneRotationTexture / BoneWeightTexture
 - Specify a 2D texture asset created with an appropriate image file.
 - Position and Rotation Precision
 - Specify the coordinates and rotation precision (8-bit or 16-bit). (8bit is sufficient to produce LOD far from the camera, as in the City sample.
 - If the mesh size is large, or if the mesh is to be placed close to the camera, consider using 16-bit. Naturally, the texture size will increase accordingly.
- Animation
 - Animation Sequences
 - Specify the animation assets to be baked into the texture. (In this case, we have registered Idle, Walk, and Run.)
 - (In this case, we registered Idle, Walk, and Run.) It is possible to output only specific frames, so adjust as necessary.

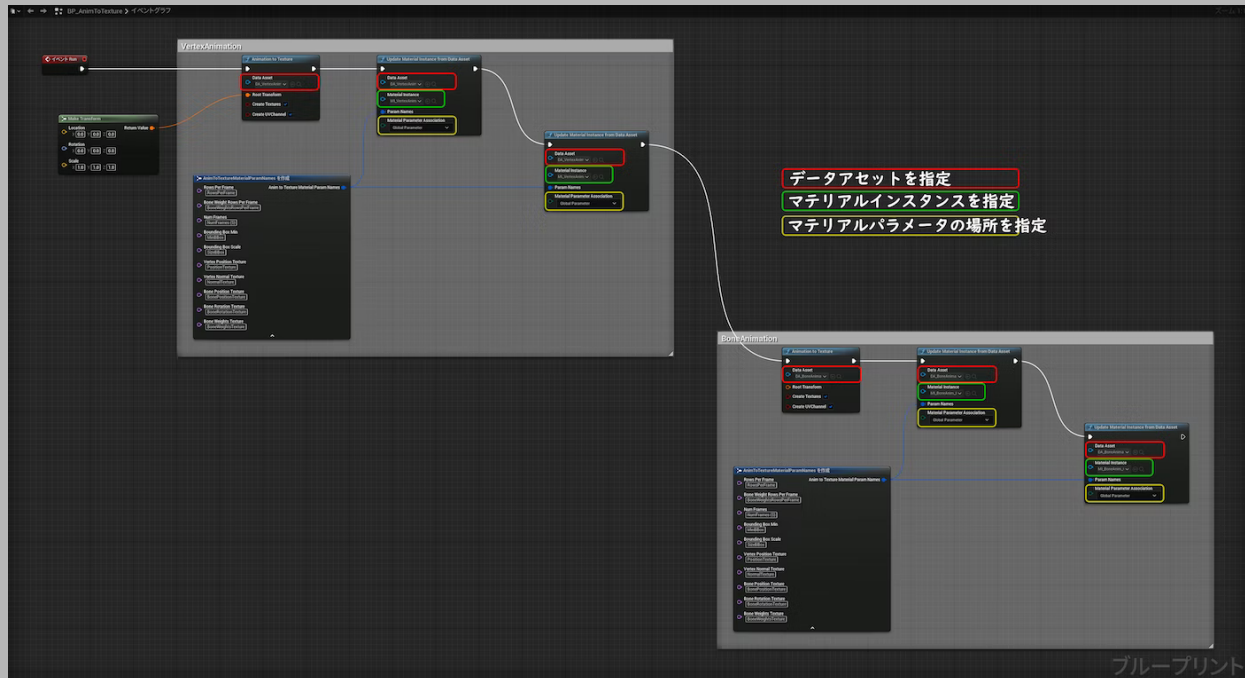
Set up the VertexAnimation data assets in a similar way.

Create a BP that performs the baking process on the texture

Create a Blueprint that will be used to burn the texture.

However, I basically just copied and pasted the assets from the AnimToTexture plugin.

Copy /AnimToTexture/Characters/Mannequin/BP_AnimToTexture to the project working folder.



The contents of the file is as shown in the figure above, with the first half being the VertexAnimation and the second half being the BoneAnimation baking process.

AnimationToTexture performs the texture baking process and updates the texture specified in the data asset.

UpdateMaterialInstanceFromDataAsset sets the parameters used for baking to the MaterialInstance.

This process automatically sets the texture and other necessary parameters in the material instance.

As shown in the figure, there are three types of parameters to be set here.

- Data Assets
 - Specify the data assets created in the previous section for VertexAnimation and BoneAnimation.
- Material Instance
 - Specify the material instance specified in the static mesh.
 - This should be done for each material instance used.
 - In the future, we will get the material instance from the StaticMesh (or SkeletalMesh) in the data asset and loop through it. It would be better to do this.
- Material Parameter Location
 - Specify the location of the material parameter to be changed
 - In this case, ML_VertexAnimation and ML_BoneAnimation are referenced as material functions, so Global Parameter is specified.

This completes the data setup.

Execution!

(There was a GIF here of the working animation. See website.)

After execution, the preview, which was broken earlier, now looks correct as shown here.

Afterword

Once you know what you are doing, it is surprisingly easy to create StaticMesh assets with animation.

However, to incorporate them into an actual workflow

- Mass production requires the creation of a convenience tool

- Although the process is automated to some extent, there are still many parts that need to be handled manually for mass production.

- Animation information textures are strongly linked to material instance parameters.

- Difficult to replace only animation information textures

- Is it okay to replace each material instance?

- Need to know the number of material instances at the baking process stage

- Is it possible to fix BP?

There are many things that could be improved, however, I think it will be a very powerful weapon if we can master it. Since Koei's Musou series, the method of animating textures with Imposter has been often used to express characters in the distance, but looking at the results, I have a feeling that it will be replaced by this method.