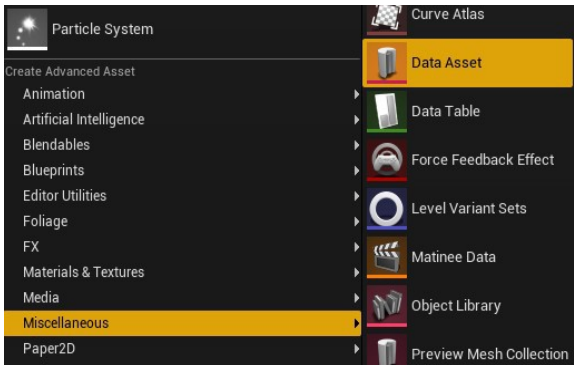
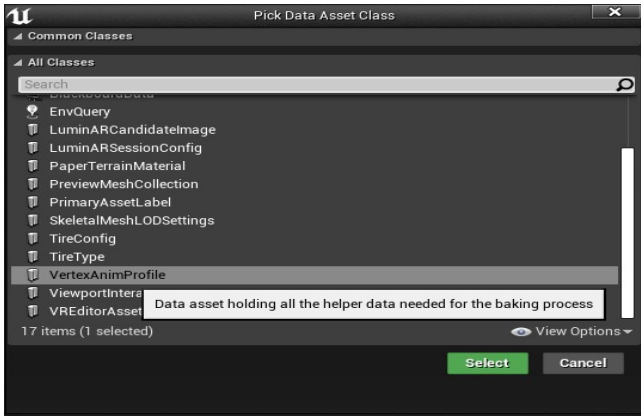
Vertex Anim Toolset V3:

Quick Start Guide:

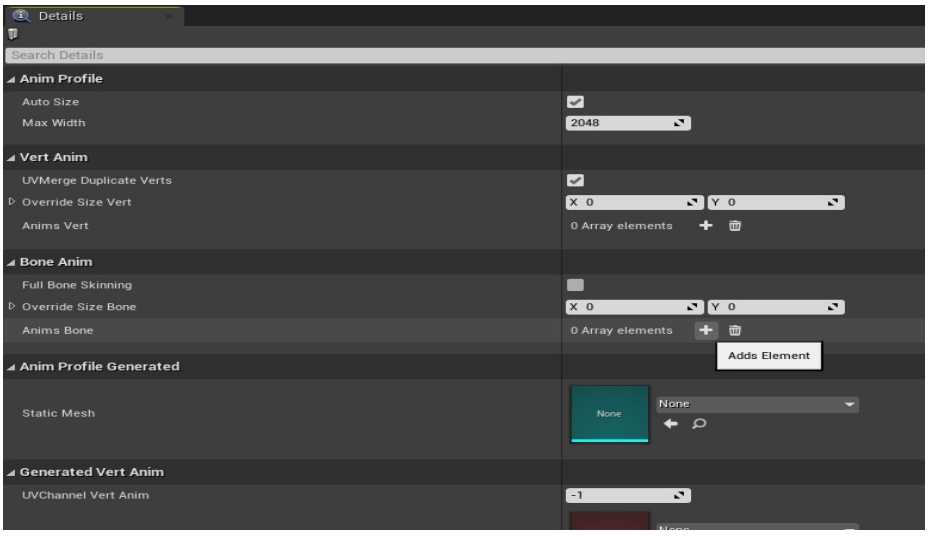
Follow these simple steps to get a result right away:

1- Create a new *Vertex Anim Profile* by right clicking in the content browser, then select *Miscellaneous->DataAsset->VertexAnimProfile*.

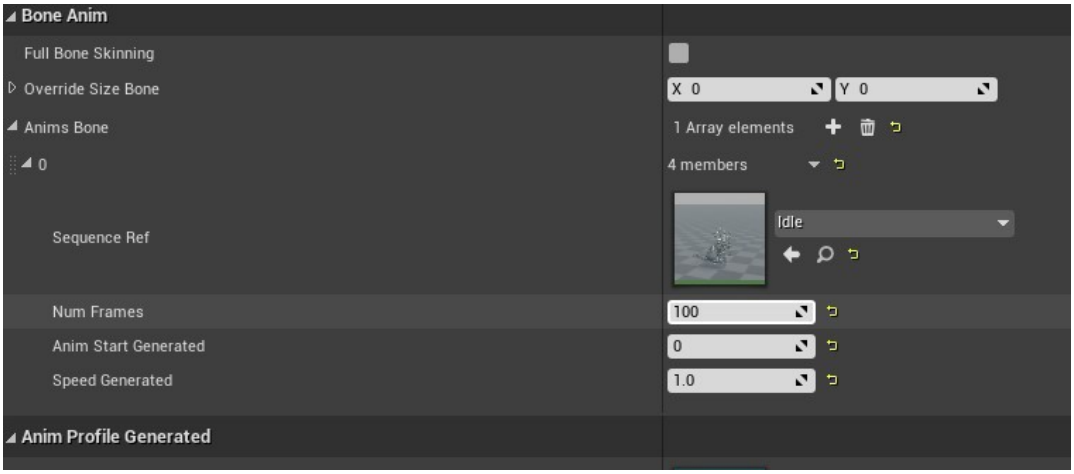




2- Open the new V*ertex Anim Profile* and go to the *Bone Anim* section and in the *Anims Bone* array variable click the *Add Element* button.

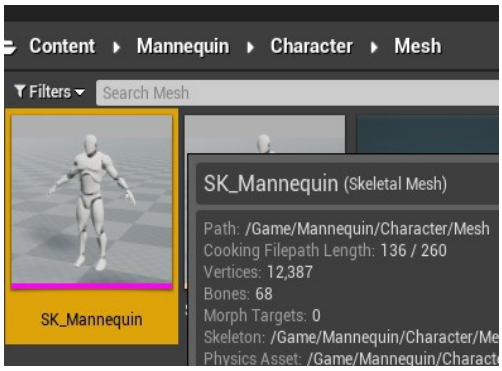


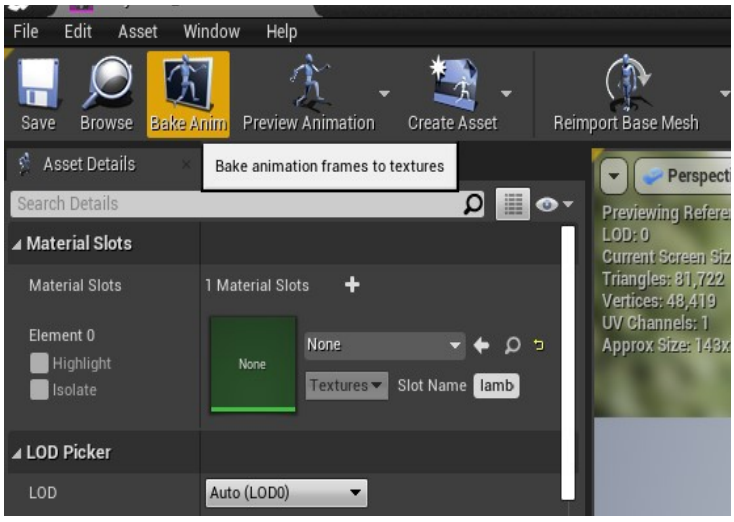
3- Open the new element of the array and on the *Sequence Ref* variable select an animation you want to bake, then set the *Num Frames* variable to something high (like 50 or 100 depending on how long your animation is).



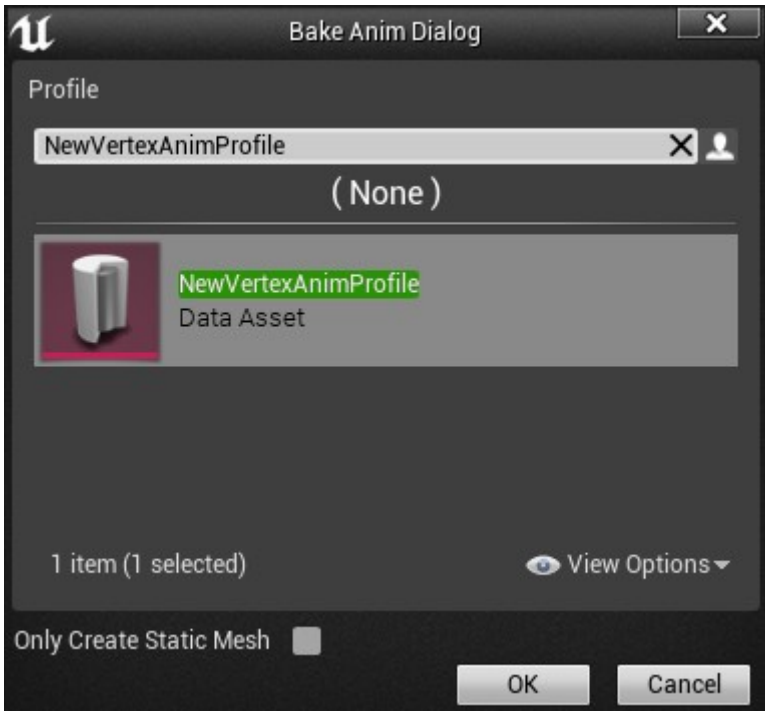
4- Save the *Vertex Anim Profile*.

5- Now open the *Skeletal Mesh* you want to bake animations for. Inside of the *Skeletal Mesh Editor* on the upper left section of the window click the *Bake Anim* button.



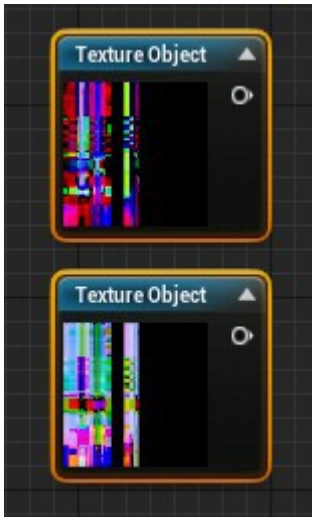


6- In the new window that opened up, select the *Vertex Anim Profile* you just created and click the *Ok* button.

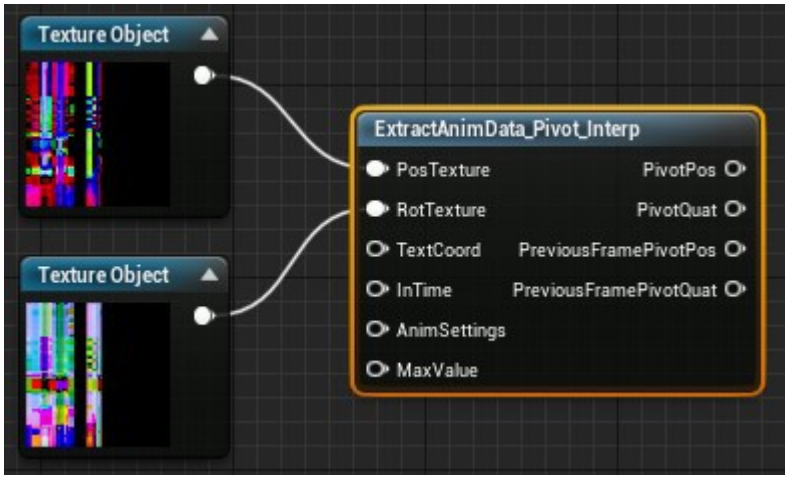


7- This process will take a couple of seconds. After its done go back to the *Vertex Anim Profile* you created and notice some of the variables have been set by the tool, keep the asset open as you'll need to know the value of these variables when creating the material.

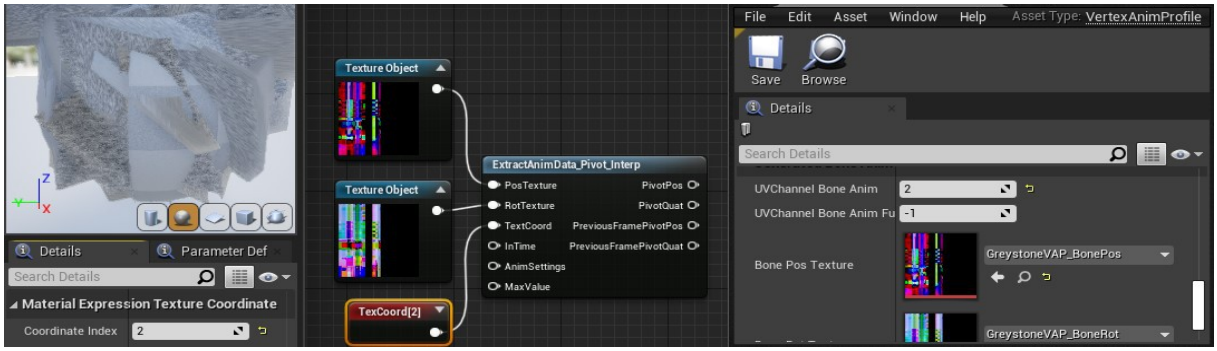
8- Now create a new material, open it up and add 2 texture objects and set them to the *Bone Pos Texture* and *Bone Rot Texture* variables inside the *Vertex Anim Profile*.



9- Now add the *ExtractAnimData_Pivot_Interp* material function and connect the previous texture objects to the *PosTexture* and *RotTexture* inputs.



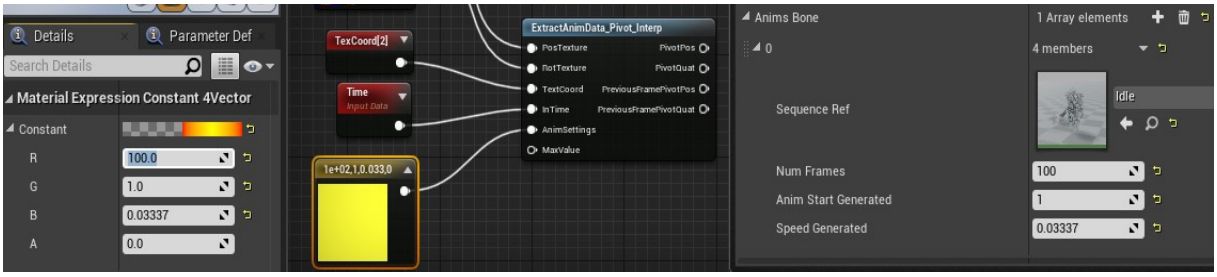
10- Add a *Texture Coordinate* material expression and set the value of its *Coordinate Index* variable to the same value of the *UV Channel Bone Anim* variable inside your Vertex Anim Profile. Then connect the *Texture Coordinate* to the *TextCoord* input of the *ExtractAnimData_Pivot_Interp* function.



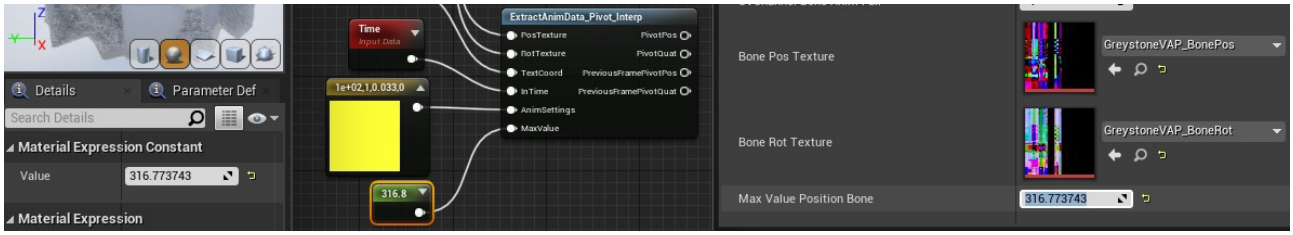
11- Add a *Time* material expression and connect it to the *InTime* input of the *ExtractAnimData_Pivot_Interp* function.

12- Add a *Constant4Vector* and set it to the values of the element added previously to the *Anims Bone* array inside your *Vertex Anim Profile*. The *RGBA* values must correspond like so: R = *Num Frames*, G = *Anim Start Generated*, B = *Speed Generated* (the A component is not used). Finally connect this to the *AnimSettings* input of the

ExtractAnimData_Pivot_Interp function.



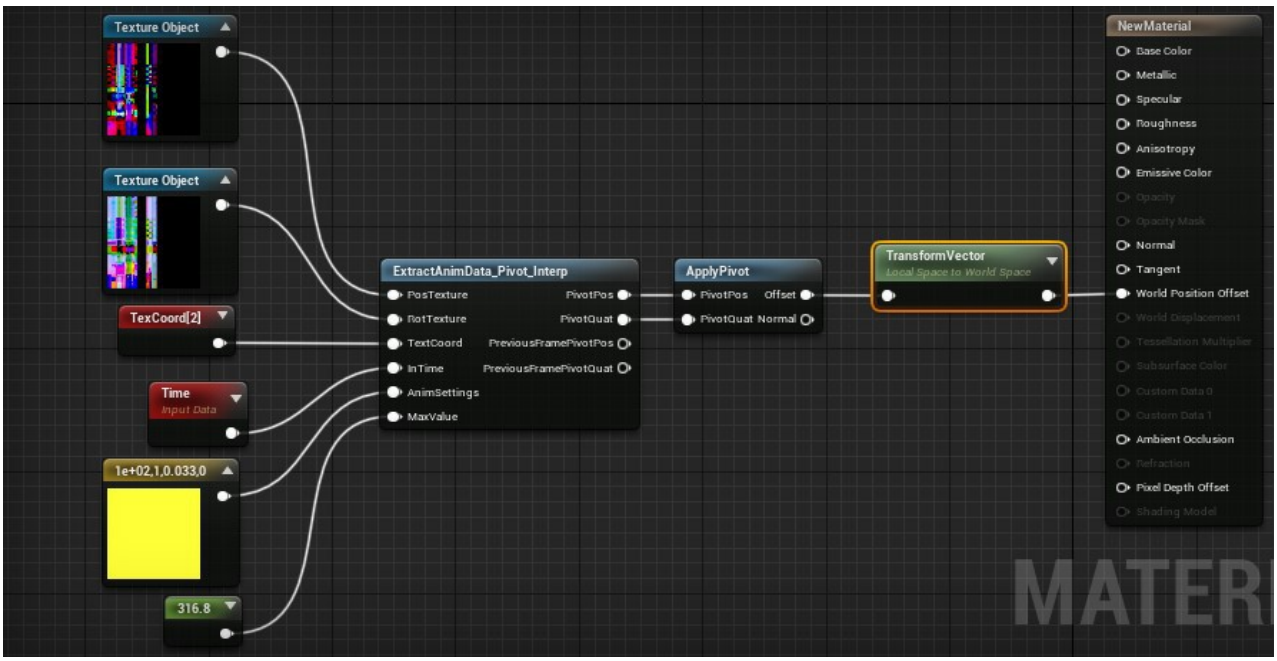
13- Add a *Constant* node and set it to the same value of the *Max Value Position Bone* variable inside the Vertex Anim Profile, and connect it to the *Max Value* input of the *ExtractAnimData_Pivot_Interp* function.



14- Add an *ApplyPivot* material function after the *ExtractAnimData_Pivot_Interp* function, connecting their corresponding *PivotPos* and *PivotQuat* pins.

15- After this add a *TransformVector* expression, setting its *Source* value to *Local Space*, and the *Destination* to *World Space* and connect the output of the *ApplyPivot* material function to it.

16- Finally connect the output of the *TransformVector* expression to the *World Position Offset* input of the material's parameters and compile the material.



17- Now go to the new *Static Mesh* that the tool created and open it. Inside the *Static Mesh Editor* go to the materials section and set every material to the new material you just created.

18- Check that the static mesh is now playing the desired animation.

1. Introduction:

This plugin allows the user to bake skeletal mesh animation into textures. This way allowing static meshes to be animated through the material shader much faster than with skeletal meshes.

Furthermore, the animation data is processed and encoded such that all LODs of a mesh use the same texture for all animations.

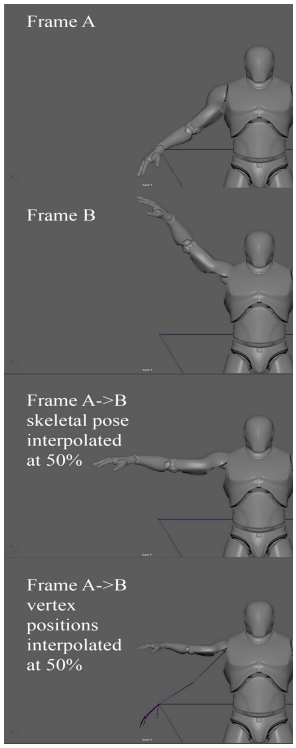
2. Bake Anim tool:

This is an In-Editor tool exposed through the “Bake Anim” button inside the Skeletal Mesh editor.

Depending on your needs, the tool can output 2 types of data baked from the animations.

As vertex offset and normal:

In this mode, it simply bakes the change in position of each vertex at every frame along with the change in normal and encodes it.



This way it is very fast to animate any mesh since all the deformation has already been computed offline and stored.

The texture size will depend heavily on the number of vertices of the skeletal mesh model you want to bake animation from.
This is because of how the plugin stores the data for each vertex along the X axis of the texture, allowing us to access the different frames of animation by simply scrolling the UVs up or down along the Y axis.

Typically this technique requires 2 texture fetches, 1 for vertex positions and 1 for vertex normals. This can go up to 4 if you are using frame interpolation. However you can go down to just 1 texture fetch for vertex positions and drop the vertex normals for maximum performance.

However, if we do not bake enough frames, using this technique can show visual artifacts related to vertex interpolation (as shown in the picture).

Because we are using vertex positions instead of full pose skinning, you'll have to make sure there's enough in-between frames to preven the model from looking bad when using frame interpolation.

<u>As delta bone position and rotation:</u>
In this mode, the tool will bake the change in position and rotation of the bones themselves, to be later used in the material to perform skinning in the vertex shader.

Compared to using vertex offset and normal, this technique is not as performant but uses a lot less data. This is because instead of storing the data for each vertex along the X axis, we are storing the data of each bone. As a result the texture size depends on the number of bones, rather than the

number of vertices.

The number of texture fetches is a minimum of 2, one for bone positions and another for bone rotations. This goes up to 4 if you are using frame interpolation. And up to 8 if you want to do the full bone skinning option.

Also, since we are adding the rotation data of the animation we have less visual artifacts from frame interpolation.

Typically you'd want to use this mode for either meshes that have a lot of vertices (40K+), or for animations that have a lot of rotation (like baked destruction).

2.1. Usage:

To use this tool, you'll need to create a Vertex Anim Profile, which is a data asset containing 2 types of variables, those which are the inputs to the Bake Anim tool, and those which are set by the tool when processing the animations (marked with the “Generated” category).

To create a new profile you must right click the content browser, then Miscellaneous->Data Asset->VertexAnimProfile.

2.2. Vertex Anim Profile variables by category:

Anim Profile:

-Auto Size: Whether to let the tool automatically detect the texture size depending on your model.

-Max Width: When Auto Size is on, this determines the maximum width the texture should be.

Vert Anim:

-UVMerge Duplicate Verts: The tool automatically checks verts that are at the same location and groups them to the same UV coordinate. This way reducing the number of vertices that need to be baked. However in case your mesh requires vertices to remain split (for example a breakable wall, or a character with dismemberment), set this to false to disable the merging of the vertices.

-Override Size Vert: If Auto Size is off, you must input the desired texture size here.

-Anims Vert: This is an array containing the data structure for each animation you wish to bake (if zero the vertex animation bake will be skipped altogether).

Inside this data structure you'll find:
-Sequence Ref: Reference to the animation sequence the tool will take the vertex deformation from.
-Num Frames: This is the number of frames that will be extracted from this sequence.
-Speed Generated: This is the speed to multiply the Time expression by to obtain the original sequence's playback speed through the material shader.
-AnimStart Generated: This is the pixel coordinate start of this animation along the Texture's Y axis.

Bone Anim:
-Full Bone Skinning: If on, the tool will bake extra skinning information, such as 4 bone indices into the UVs, along the skin weights into the model's vertex color. Allowing you to do full vertex skinning on the material shader.

-Override Size Vert: If Auto Size is off, you must input the desired texture size here.

-Anims Bone: Array of animations to bake bone animation from (if zero the bone animation bake will be skipped altogether).

Anim Profile Generated:
-Static Mesh: Reference to the static mesh created by the bake tool. If set this mesh will be overwriten everytime the bake tool is used.

Generated Vert Anim:
-UV Channel Vert Anim: Texture coordinate index where the UVs for extracting the data from the textures are stored. If there was no vertex animation to bake this will be -1 since the tool also skips adding the UV channel for vertex animation.

-Offsets Texture: Reference to the created texture asset.

-Normals Texture: Reference to the created texture asset.

-Rows Per Frame Vert:

-Max Value offset Vert:

Generated Bone Anim:
-UV Channel Bone Anim: Texture coordinate index where the UVs for extracting the data from the textures are stored. If there was no bone animation to bake this will be -1 since the tool also skips adding the UV channel for bone animation.

-Bone Pos Texture: Reference to the created texture asset.
-Bone Rot Texture: Reference to the created texture asset.

-Max Value Position Bone:

Material Functions:

The plugin incorporates various material functions for decoding and using the data baked into the textures. You'll find all these material functions inside the Content folder of the plugin (make sure inside the “View Options” of the content browser that “Show plugin Content” is checked).

Here's a brief explanation of each Material Function:

-Apply Pivot: Takes the animation data of one single bone and computes the corrseponding vertex offset and normal.

-AxisBasedTransform: Takes a local space vector and transforms it into world space ignoring the scale of the object completly.

-DecodeQuat: Takes the encoded RGBA vector from a HDR Rot texture and decodes it to obtain the corresponding quaternion.

-DecodeVector: Takes the encoded RGBA vector from a LDR Pos texture along a Max Value and decodes it to obtain the corresponding vector.

-DecodeVectorHDR: Takes the encoded RGBA vector from a HDR Pos

texture along a Max Value and decodes it to obtain the corresponding vector.

-ExtractAnimData_Vert: function for taking the baked vertex animation data and outputting the vertex offset and normal offset in local space.

-ExtractAnimData_Vert_Interp: function for taking the baked vertex animation data and outputting the vertex offset and normal offset in local space. With additional functionality to interpolate between frames for a smoother look and allow for custom motion blur.

-ExtractAnimData_Pivot: function for taking the baked bone data and outputting a pivot position and quaternion to apply to the vertex in local space.

-ExtractAnimData_Pivot_Interp: function for taking the baked bone data and outputting a pivot position and quaternion to apply to the vertex in local space. With additional functionality to interpolate between frames for a smoother look and allow for custom motion blur.

-ExtractAnimData_Skin: function for taking the baked bone data and outputting the vertex offset and normal in local space.

-MultiplyQuats: function to combine a quaternion with another quaternion.

-QuatNormalize: function to compute the normalized version of a quaternion.

-QuatRotateVector: rotate a vector by this quaternion.

-QuatUnrotateVector: inverse rotate a vector by this quaternion.

-SlerpQuats: Computes the interpolated version from one quaternion to another.

-TransformUniformScale: function for transforming a local space vertex offset and normal offset into world space ignoring the scale of the object completly.

Final Notes:

-With baked bone animation you can have different meshes use the same texture as long as they have the same skeleton.

-With baked bone animation the tool bakes the reference pose into the first row of pixel values.