User Documentation

Introduction

Puppet3D is a tool to Rig and Animate your characters in Unity.

Getting Started

To launch Puppet3D go to Window>Puppet3D

This will open the following UI:

There buttons at the top Tab between different sections:

AutoRig - This is where you can use the **AutoRig** and **Mod Rig** feature
Skeleton, Rigging, & Skinning - These Panels are used for **Manual Rigging**
Animation - A panel with some tools for animating
About - Version Number (Quote this in any bug reports)

In general there are 3 ways of working with Puppet3D; **AutoRig**, **Mod Rig**,and **Manual Rigging**.

Mod Rig - If you have a humanoid character that already has bones, and you want to modify the animation, or make some new animations see this section.

AutoRig - If you have a biped model and you want to rig it quickly see this section.

Manual Rigging - If you want to create bones, controls and skin your character yourself you want this section.

AutoRig

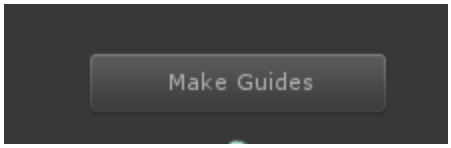
Want to quickly rig your character model? This is the section for you.

Step by Step

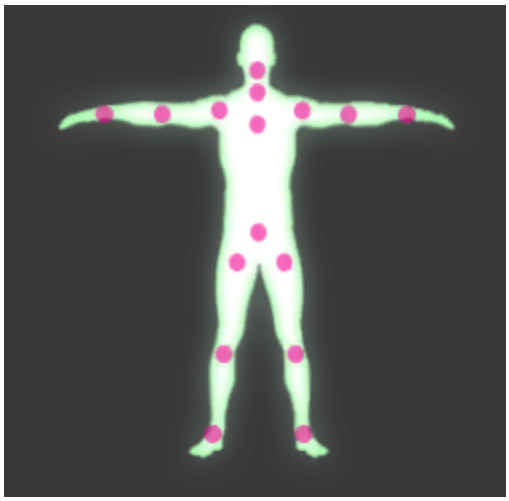
1. Make sure your character is in a Tpose facing down Z direction. Legs should be apart slightly.

2. Drag in your model from the project into the hierarchy.

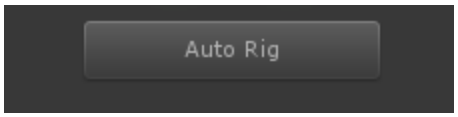
3. Select it and click Make Guides (Make you've selected the gameObject with the MeshRenderer)



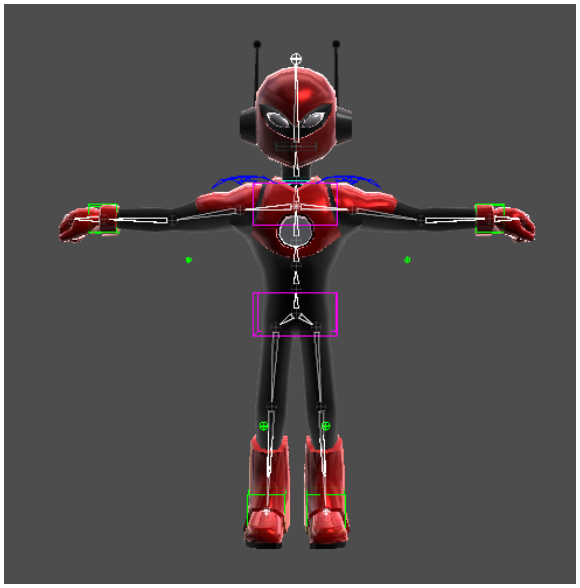
4. You should see some guides to move. Move each of these to the correct spot in your character (use the following image a reference for where they should go).
    - The handles should snap to the centre of the character. If this doesnt work well, then you can uncheck this in the inspector and move it manually.
    - The left and right handles mirror each other. If you uncheck this in the inspector you can move them independently.



5. Now click "AutoRig"



6. Your character should be rigged.



7. Now you may want to paint the skin weights a bit, see Painting Weights

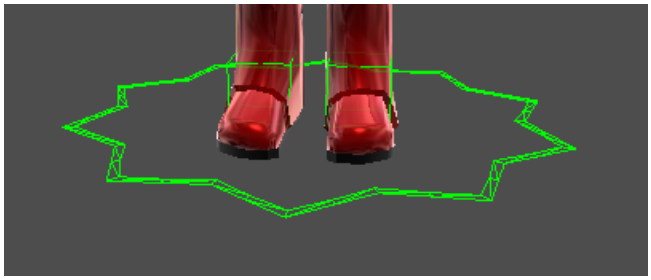
NB It's a good idea to put your character on a locked layer to stop accidental selection whilst moving the handles.

Using The Rig

The rig is made up of controls, and bones. You can hide and show these global control. The controls are the things you want to animate - the bones are being driven by the controls. The character itself is skinned to the bones.

The rig has been made to fit the local proportions of your character. That way animations will mostly be transferable between different rigs. (There may need a little adjustment here and there, but they should mostly work).

Global Control



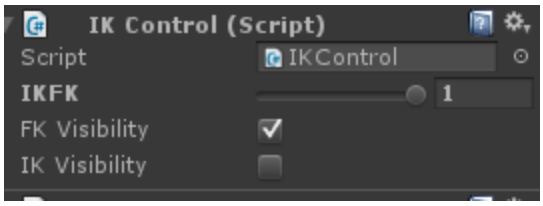
To move the whole character around you can use the global_CTRL. This is the master gameObject which runs all the other controls, and turns on and off their visibility. You tend to have the animator on this gameObject, or you can have it on a gameObject above it.

NB Whilst you're animating it's recommended to break the prefab instance of your character - this will make selecting controls much easier.

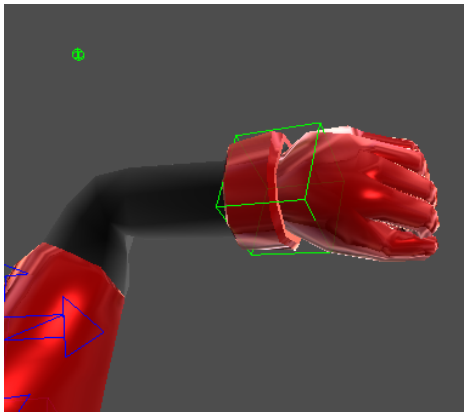
IK/FK Arms & Legs

IK (Inverse Kinematics) is where you move the control, and the limb moves to meet it.
FK (Forward Kinematics) is where you rotate each of the limb joints to the desired position manually.

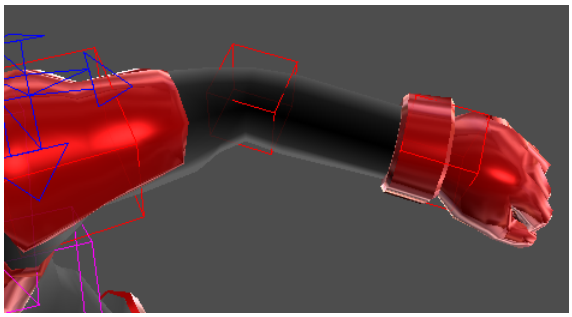
Depending on what you want to do or how you like to animate you may want to use either method. This rigs allows you to use both and animate between them - this is called IK/FK blending. This blend value (IKFK) can be accessed on the IK control:



The IK is made up of the IK control (green cube), and pole control(green sphere). The IK control tells the arm where it wants to end, and the pole control tells the elbow which way to aim.



The FK controls are made of three red boxes that you rotate to whichever direction you want the arm to be. (You can toggle their visibility in the hierarchy on IK controls FK visibility checkbox.)

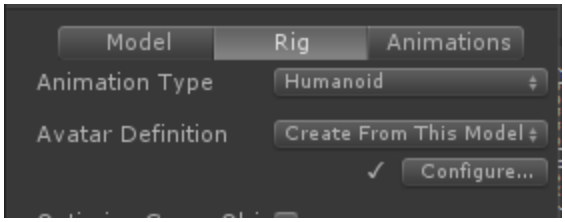


Mod Rig

If you have a biped character that already has bones, you can quickly add a mod rig to it. This will allow you to easily animate the character, as well as modify any animations that are already on it.

Step By Step

1. Select the character in the project and in the inspector, make sure it's set to humanoid in the Rig tab.



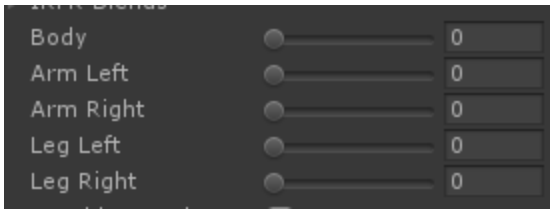
2. Hit Configure to create the avatar. This will define all the bone names.
3. Now drag your character into the hierarchy, select the object with animator (needs the avatar applied) and click “Create Mod Rig”
4. Your character should now be rigged!

NB. If you don’t have all the bones defined, it may not be able to create the rig, or it may not have the best results.

Using The Mod Rig

The mod rig works in much the same way as an AutoRig. The one extra feature is the ability to blend in and out of the premade animation.

If your character has any animations on it prior to the mod-rig, you can can blend these on and off from the global_CTRL.

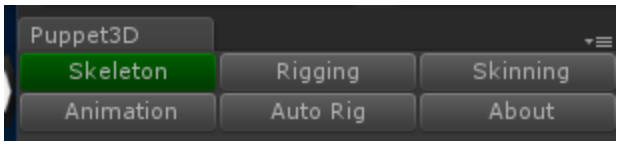


You can blend of the body, and limbs each individually. If they are all on 1 the character will move with his original animations. When set to 0 they will use the mod-rig controls. You can animate these values - useful if you want to adjust a certain part of the animation. For example if you want to make his arm move to a specific spot, you can keyframe on the arm, and place it to where you want it, animate it and then key it back.

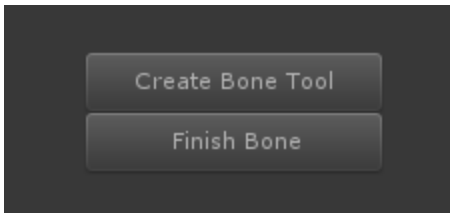
Manual Rigging

With Puppet3D you can rig pretty much anything. It’s all about the skeleton, skinning and controls.

Skeleton



Bone Tool

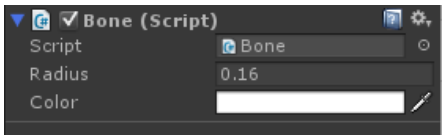


Bones are visual representations of your characters articulated hierarchy.
To start creating bones click the “Create Bone Tool”.

Left Click in the scene view you can draw bones (from joint to joint).
Pressing Enter, or clicking the “Finish Bone” button will exit the tool.
Hold Shift to move the current bone.
Hold Ctrl and left click to select another bone in the scene view.
Right Click to end the current bone chain without exiting the tool.
You can edit the bones by reentering the bone tool.

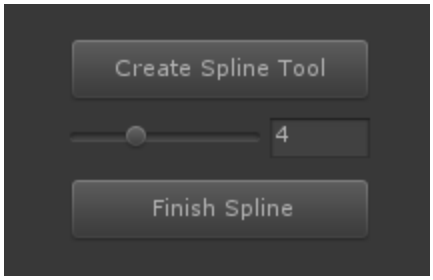
NB If the bones are rigged you should not move the bone positions or delete them. Instead you would need to first delete the controls and remake them afterwards.

You can change the color and radius of each bone in the inspector.



NB. Add a mesh collider to your character so that the bones will project into the centre of your character.

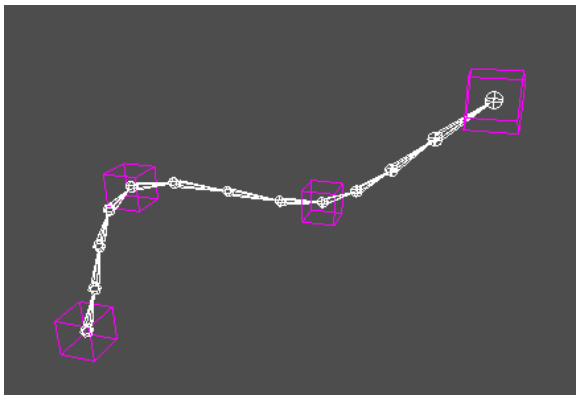
Spline Tool



You can make long multi-bone chains using the Spline Tool. These are perfect for tails, tentacles, long bend pipes, and also for the spine of your character.
This tool also makes the spline controls themselves. (If you only want the chain, just delete the controls after creating the bones)

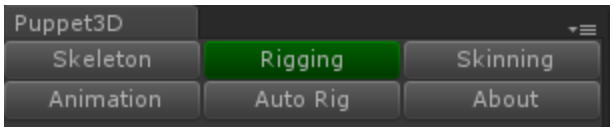
Click “Create Spline Tool” to start creating the spline bones. You can set the number of bones between each spline control from the slider.

Left Click in the scene to start drawing the spline controls. You need to make a minimum of 2 controls and then press Enter or “Finish Spline” button to complete the spline. It should create the spline bones like this:



Hold Shift to move the current control.
Hold Ctrl and left click to select another control in the scene view.

Rigging

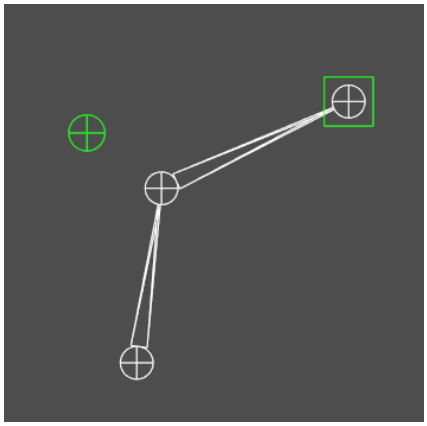


Bones aren't very animator friendly. To make animating more intuitive, you can make a controls that move the bones. These controls will have their local transforms all clean making the animation curves all start from zero.

IK Controls & Limb Controls

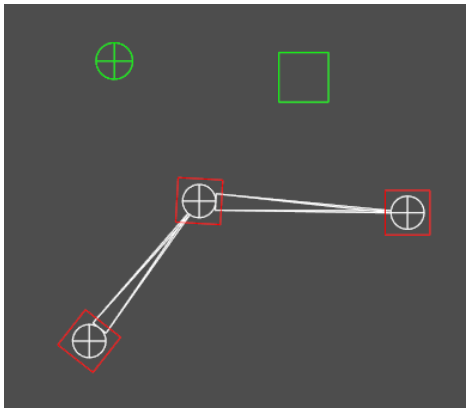
These are used for limbs such as arms or legs. (Limb controls are similar to IK controls, with the addition of IK/FK blending).

To create an IK control or Limb control, select the end joint of a 3 joint chain and click "Create IK Control" or "Create Limb Control". For example if it was a leg, you could have a 3 joint chain like this: Thigh>Knee>Foot - you would select the Foot joint and create the control.
It will create the IK control (in this case its call Foot_CTRL). This is the control you can use to animate. Notice that it has a parent GameObject (Foot_CTRL_GRP). This keeps the controls transforms in local space. This shouldn't be animated, but can be used to change the default starting place for the control. If you want to parent this IK control to another control, parent this gameObject instead.



NB Try not to parent controls to bones, this can cause a cycle error as the the control influences the bone which influences the control ad infinitum.

Limb controls have the added feature of IK/FK blending. (See “Using The Rig” for an explanation on what this does). There are two visibility checkboxes that show/hide the IK and FK bones themselves. If you want to reparent the arm you would need to show these and reparent them as well.

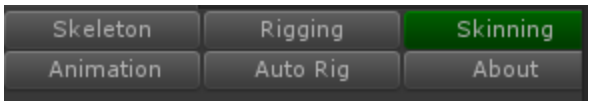


Parent & Orient Controls

These directly control the bones. Parent controls influence both position and rotation, whereas Orient controls only influence the rotation.

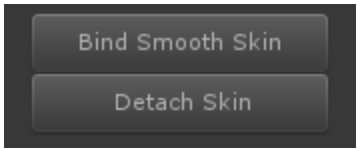
NB When parenting these controls to each other make sure you parent their parent GameObject so they maintain their local transforms

Skinning

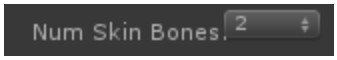


Skinning is where you make the bones influence your character. Essentially it will be making the meshRenderer into a skinnedMeshRenderer setting up all the bone and weight assignments.

To skin your character to the bones, select the gameObject with the meshRenderer and all the bones you want to skin it to and click “Bind Smooth Skin”.
You can detach the model from the rig by selecting the gameObject with skinnedMeshRenderer and clicking “Detach Skin”



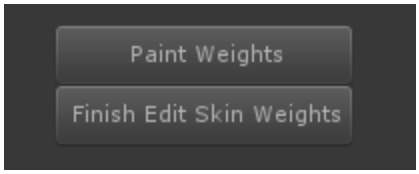
NB You currently have a choice of 1 or 2 bone skinning. 4 bone skinning isn't presently supported.

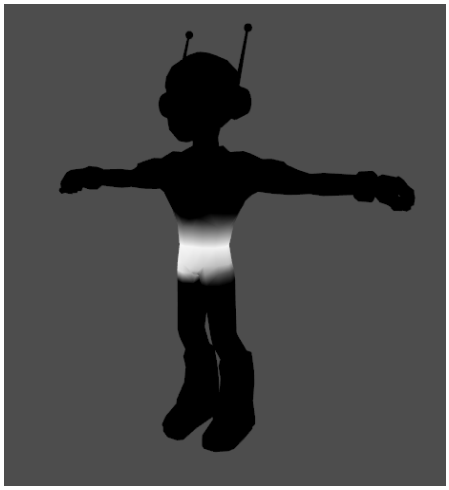


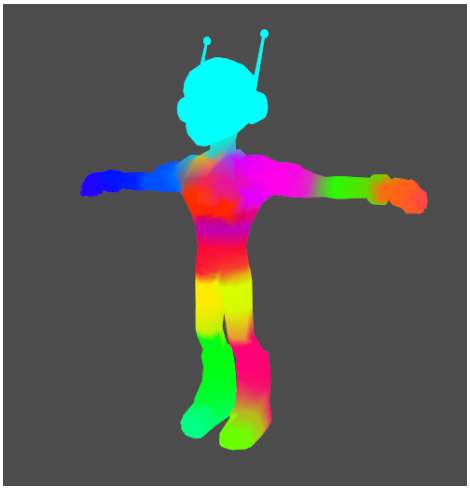
Skinning Algorithms

There are two types of skinning algorithms which affect what your default skin weights will be; Closest Point & Voxel. Set this before you click "Bind Smooth Skin".
Closest Point is quicker, but it just try to find the nearest bone to each vert, and then blend it with the bone above it.
Voxel skinning will actually create a voxel mesh out of your character and spread the weights through this mesh. This can be more accurate and will stop weights from spilling from close but unconnect places (eg from one leg to another).
If you tick "keep voxels" you can see a copy of the voxels that were used to skin your mesh. If the voxels are to big, you can change the voxel resolution (higher values give more smaller voxels). You can also change the amount of smoothing you get between voxels. This will affect how "bendy" the default weights will appear.

Painting Weights





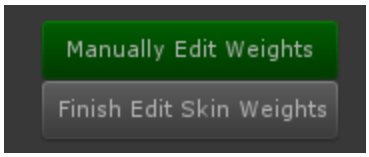


Each vertex of your character has weights to 1 or 2 bones. You can change these weights and their bone assignments by using the paint weights tool.
Select the gameObject with skinnedMeshRenderer and click "Paint Weights". (Make sure it has a meshCollider component attached).

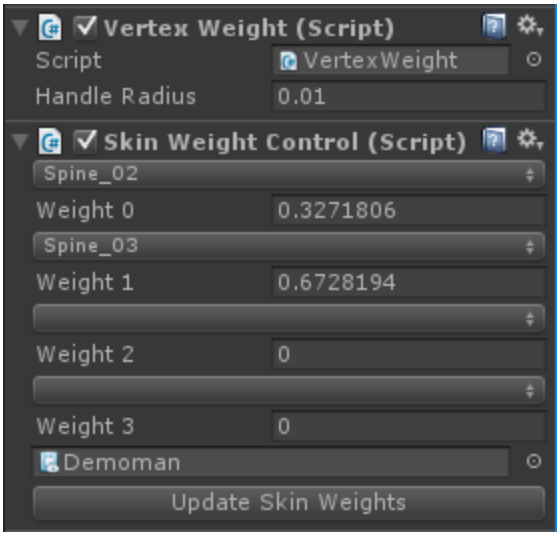
Now you can select a bone to paint (in the hierarchy).
Then with Left click you can paint with weights for this bone.
Holding Ctrl with Left Click removes the weights for this bone.
Holding Shift and Left Click smooths the weights for this bone.
You can change the radius either with the UI slider or by holding B and moving the mouse left and right.
You can change the strength of the painting either with UI slider or by holding N and moving the mouse left and right.

There are two types of painting view:
Black & White - shows you the weights of the current selected bone.
Colored - shows the weights of all the bones at once, each with a different color.

Manually Edit Skin Weights

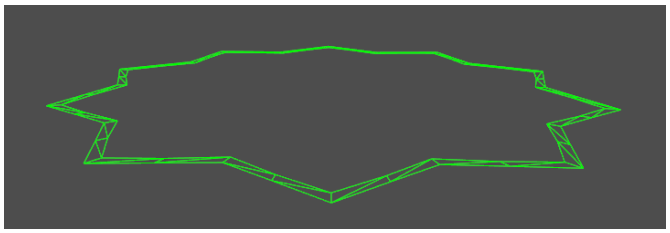


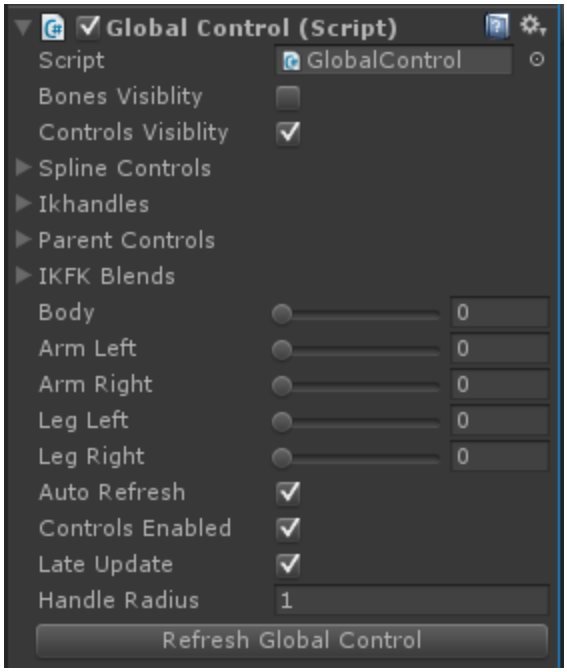
You can manually edit each skin weight if you click the Paint Weights button twice. (It toggles between paint weights and manual edit mode).
In this mode you can select each vertex and set each of the bones and weights in the inspector.
If you want to change multiple vertices at once you need to click the “Update Skin Weights” button with all the vertex selected.



To exit the paint weights tool press the “Finish Edit Skin Weights” button.

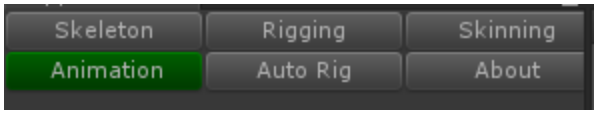
Global Control



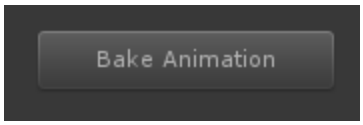


This is the master control which runs all the other control scripts. Everything should be parented to the global_CTRL. Whenever you create anything new Puppet3D will parent it to this control.
You can disable all the controls with the enable controls checkbox.
You can toggle the visibility of the bones and controls from this gameObject.
The Body/Arm/Leg sliders are used for Mod Rigs.
If you make any changes to the number of controls you can click the “Refresh Global Control” button to reset all the control assignments on the global_CTRL.

Animation



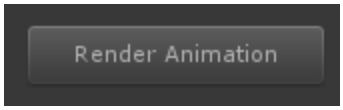
Bake Animation



You may want to remove all the Puppet3D control scripts from the game. They aren't very expensive, but they do take up a little bit of CPU. To do this you can bake all the animations from the controls onto the bones.

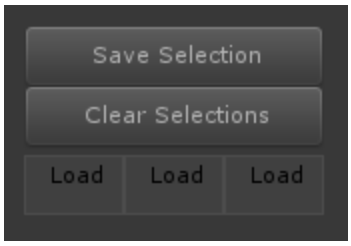
To do this make sure your character's rig is in the hierarchy and all the animations you want to bake are attached to the animator. Then click "Bake Animation" and it will create a folder with all the baked animations in Puppet3D>Animation>Baked
You then can remove all the non-baked animaitons in your animator and replace them with the baked versions.

Render Animation



Puppet3D comes with an animation Renderer. Make sure you have a camera in the scene set to tag MainCamera. Click "Render Animation" and select the folder you want it to save into and the name. It will play the game and make a png sequence of the animation into that folder.

Pose Saving/Loading



You can store poses and selections temporarily to help you whilst animating.
To use, select all the controls for your pose/selection and click "Save Selection". It will add a "Load" Icon to the UI. You can now Load the Pose by right clicking on the Load word and click "Load Pose".
You can also do the following:
Select Objects - This selects all the controls for this pose.
Remove Selection - This removes the current selection from this poses controls.
Append Selection - Adds the current selection to this poses selection.
Store Pose - Stores the current pose state of this poses' controls.