AvatarPlay

Version 0.10

User's Guide

Contents

Overview

AvatarPlay is an seperated animation system in unity. It contains an open source low level animation engine, and clean animation clip format. It's easy to add your own animation features or functionality base on AvatarPlay framework. It's a completely avatar-base animation system with comprehensive support for animation re-targeting, including non-humanoid or nonstandard humanoid models.
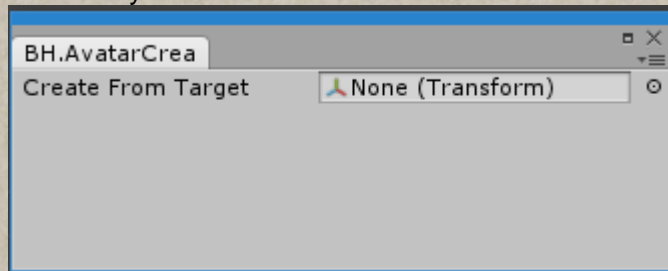
Avatar

Concept

Avatar define the bone structure and T-Pose of a certain type of animation target. When create animation clips from AvatarPlay, you must specify an avatar as the criterion reference. Before running animation clips for a model, you must create a mapped from the avatar which was the animation clips created from.
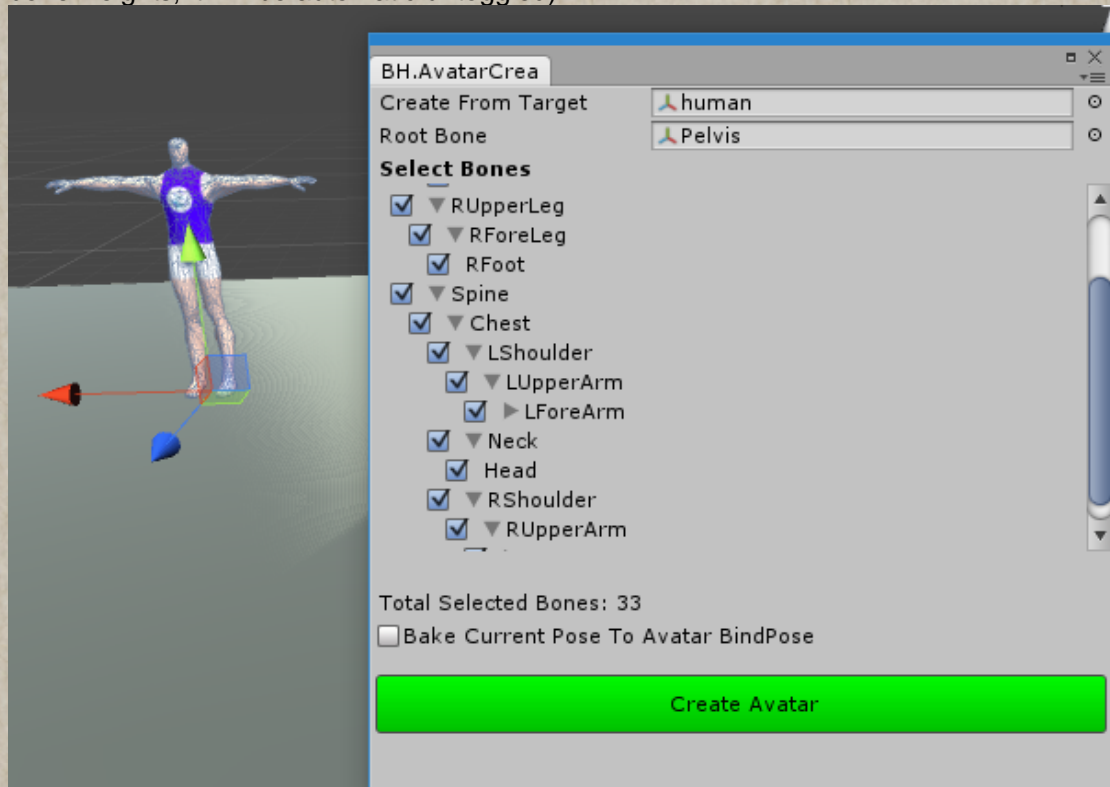
Create Avatar

1.Open Create Avatar window from Assets -> Create -> AvatarPlay menu or Window -> AvatarPlay menu.

2.Select a model transform as the avatar source target.
E.g. If you are creating a humanoid avatar, you should specify a typical human model.

3.Specify the root bone of the skeleton (generally it will has been assigned automatic), and select bones which you want to export to avatar (if any bone branch has no skinning bone weights, it will be automatic untoggled).


4.Toggle "Bake Current Pose..." if you want to bake current target pose to avatar, or the target model's bind-pose will be apply to avatar.

5.Click "Create Avatar" when you are ready, then the save-file panel will be opened. The default saving path should be in AvatarPlay\avatars directory.

Configure Avatar

1.Select avatar object in the Project Window, then you will see the summary of the avatar in Inspector Window, click "Configure Avatar" button to configure avatar.
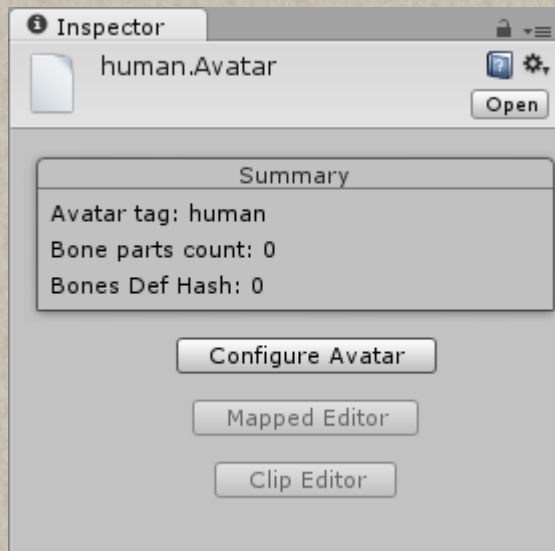
2.When entering "Configure Avatar" mode, a temporary new scene will be created, the Inspector Window will be locked. You will see the avatar model in the scene view, and configuring options in the Inspector Window.
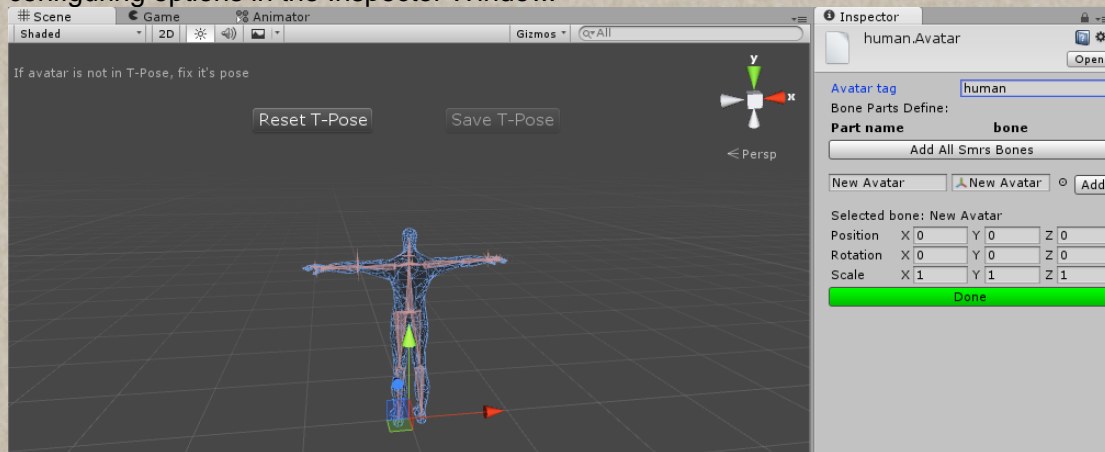
3.Define bone parts: Click "Add All Smrs Bones" can automatic add all bones from the Skin Mesh Renderer of the avatar. Select bone in Hierarchy Window or in Scene Window, then click "Add" button in the Inspector Window can add a single bone. Click "Del" button to remove bone from Bone Parts Definition.
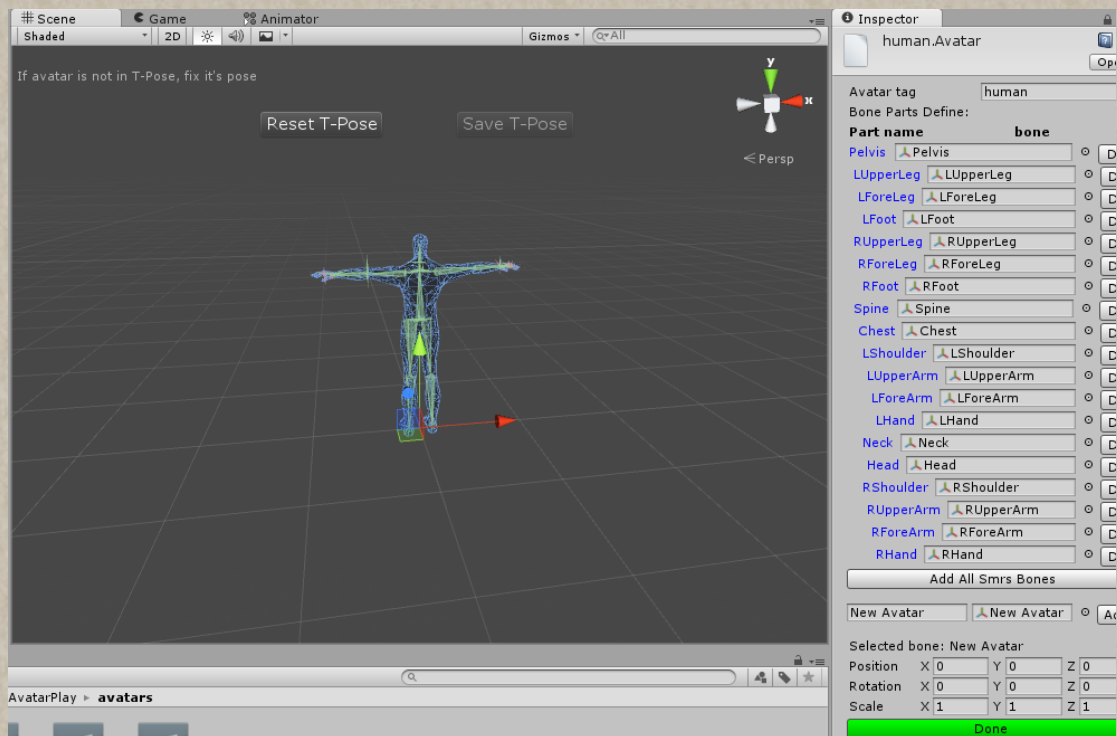
4.Define the T-Pose of the avatar: T-Pose is a standard reference pose for adjusting pose with certain model when you create Avatar Mapped. Generally you can let it to be the bind-pose of the source skinning model. You can change the avatar pose in the Scene View by changing the transform position or rotation of a bone which have been added to bone parts definition.

Avatar Tag

Avatar Tag is the unique identifier of the avatar. Avatar mapped objects and Avatar Animation Clip objects associate with Avatar with this tag. You can modify this tag through Avatar Configuration, but all association of the objects connected to the previous tag will be lost.
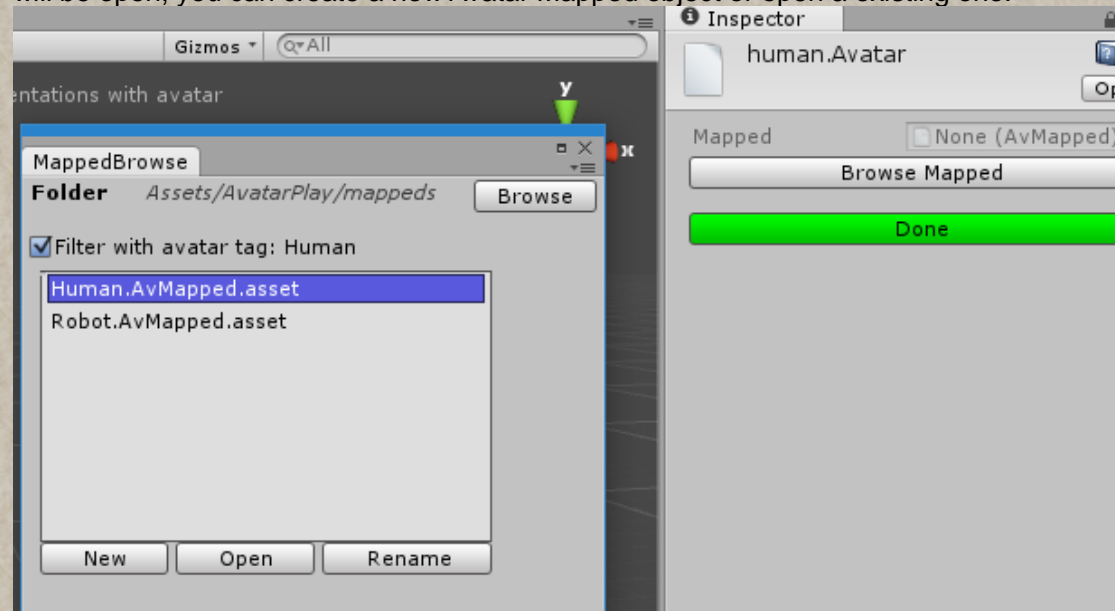
Avatar Mapped

Concept

An Avatar Mapped object store the bones mapping result from an Avatar to a Model. Avatar Mapped is the bridge between avatar and model, in this way, the animation clips created through an Avatar can be applied to models mapped to that Avatar.
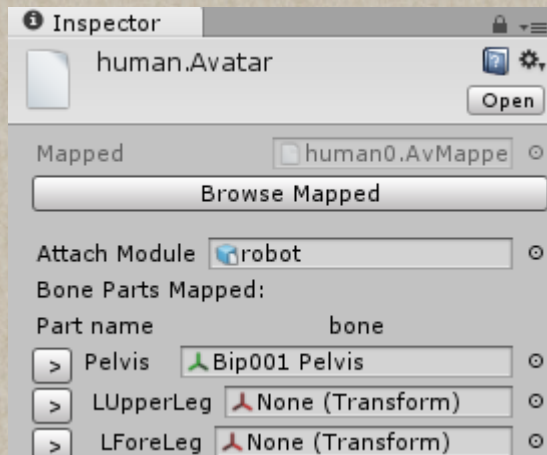
Create & Edit Mapped

1.By clicking "Mapped Editor" button from an Avatar object's Inspector Window, or "Edit Mapped" button from Avatar Mapped object's Inspector Window, you can enter Mapped Editing Mode.
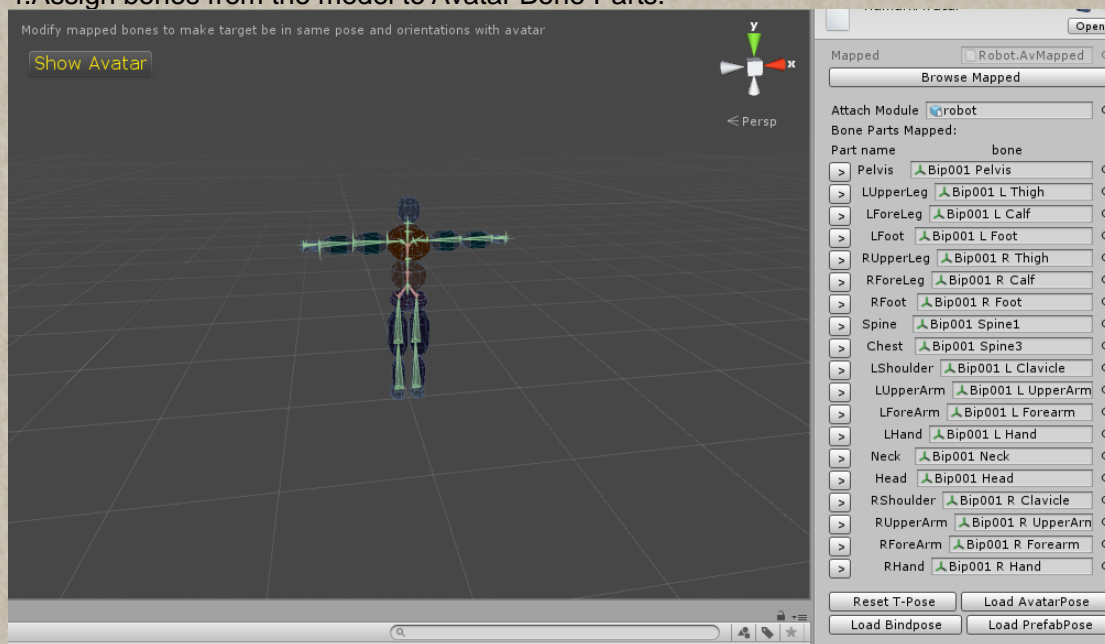
2.Click "Browse Mapped" button from the Inspector Window, then the Browser Window will be open, you can create a new Avatar Mapped object or open a existing one.
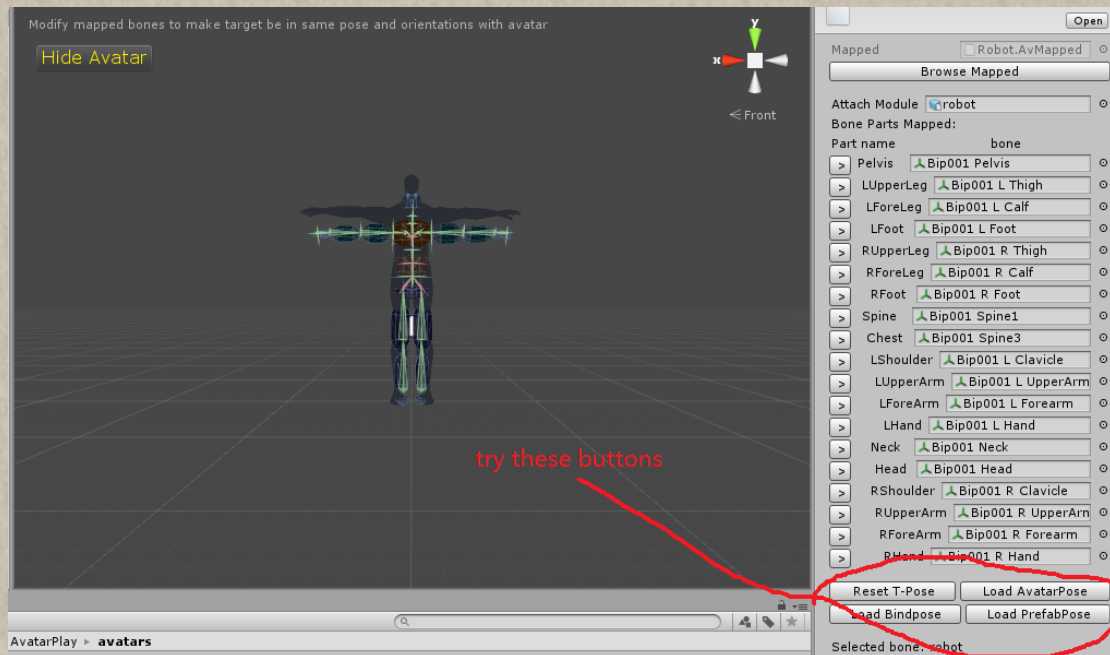

3.After creating a new Avatar Mapped object, you must specify the model to be mapped, through the "Attach Module" field.

4.Assign bones from the model to Avatar Bone Parts.

5.Adjust the pose of the model showing in the Scene View to Avatar T-Pose.

Avatar Animation Clip

Concept

An Avatar Animation Clip object stores bones (from Avatar) transformation data in the form of key frames and time line. In AvatarPlay system, you always create clips for an Avatar instead of a Model, but you can play the clips for an Avatar on any model which has been mapped to that avatar.

Create & Edit Clip

1.By clicking "Clip Editor" button from an Avatar object's Inspector Window, or "Edit Mapped" button from Avatar Mapped object's Inspector Window, you can enter Clips Editing Mode.

2.Click "Browse Clip" button from the Inspector Window, then the Browser Window will be open, you can create a new Avatar Animation Clip object or open a existing one.
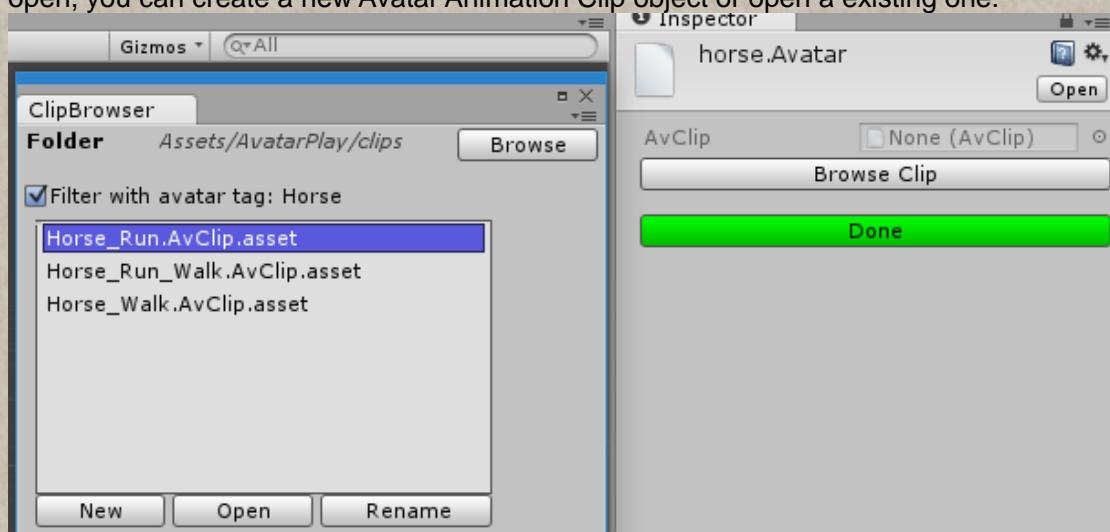
3.Now you can see the Avatar in the scene view, and Key Frame Editing in the Inspector Window.
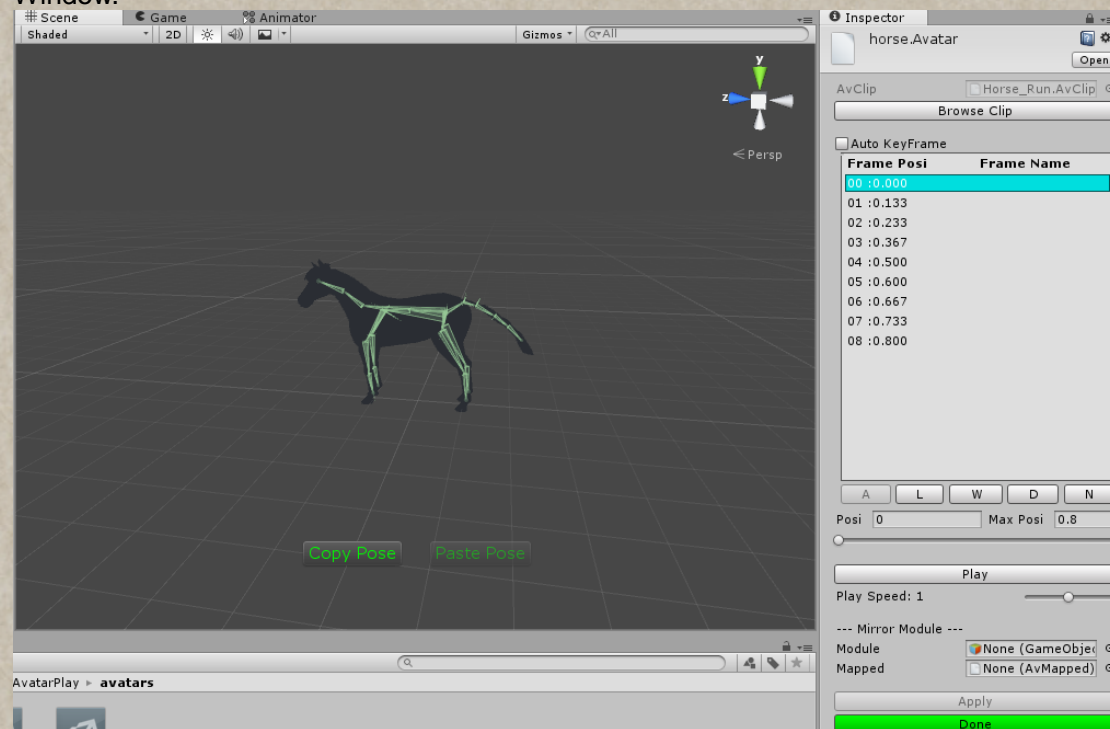
4.UI Introduction:

A: Add a key frame with current pose to current clip position.
L: Load pose from selected key frame.
W: Write current pose to selected key frame.
D: Delete selected key frame.
N: Naming selected key frame.

Posi: Current clip position.
Max Posi: Max length of the clip.
Play: Run the clip.

Auto KeyFrame: Activate auto key frame mode, then "Load Pose" and "Write Pose" will be executed automatically.

Copy Pose: Copy current pose, but the root motion change will not be copied.
Paste Pose: Paste latest copied pose, excluding root motion change.

Mirror Module:A model as reference while editing clip. Since you specify a model, you also need to specify the mapped. The pose of Avatar and Mirror Module will be kept in sync.

Bring your models to life

Create Clip Set
A Clip Set contain a collection of clips and configurations to them. You can create a Clip Set by Create->AvatarPlay->AvClipSet menu. Then add clips to the clip set and do some simple options.
Notice: All clips added to a Clip Set must be attached with a same Avatar Tag.
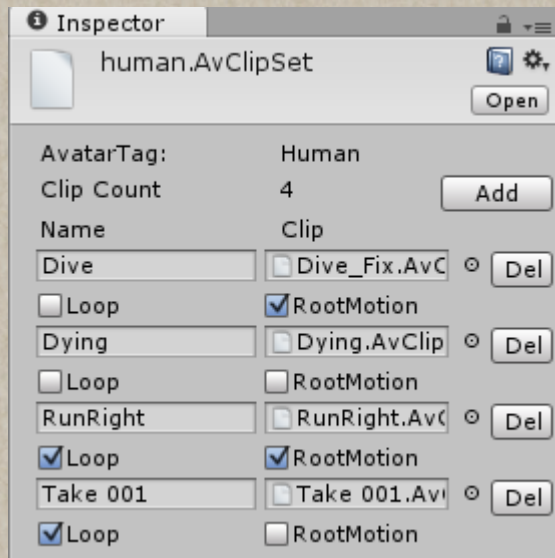
Add Avplayer Component to Model

Avplayer component is the playing controller of AvatarPlay animation clips, like Animator or Animation component of unity.

You must specify a Mapped and a Clip Set before playing clips, and the Mapped and Clip Set must be attached with a same Avatar Tag.

If you specify the default clip, the clip will be played automatic while the game is running and no other clip is specified to play currently.

You can test clips in edit mode by the buttons of "Clip List" in the Inspector View of Avplayer component.
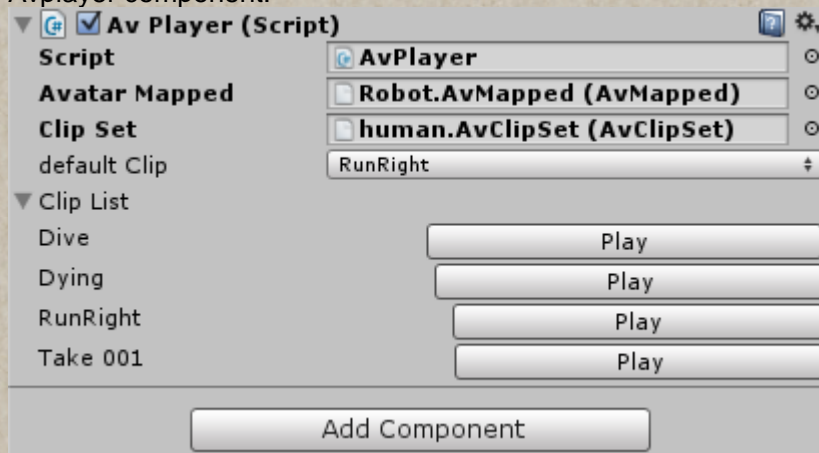
Control Animations from Code

For controlling animations from code, you have to get the Avplayer component first, by using GetComponent function. It's recommend to skim through these class in code: AvPlayer, AvPlayerExec, RuntimeAvClip.