IK Helper Tool 1.1

Game Engine: Unity

Kevin Iglesias - www.keviniglesias.com

Contact Support:
support@keviniglesias.com

1.1 Online Documentation

1.0 Documentation (old version)

INDEX

Version Changes

Version 1.1

- Added new custom inspector with better stability, performance and compatibility with Undo and Redo commands.

- Added Edit Mode that freezes animation for easier setup of IK attachment Transforms while in Play Mode.

- Improved performance of the main script.

- Added new scripts to work as State Machine Behaviours.

- Added smooth transitions between IKs.

- Now you can use the default animation in a Sequence IK (previously called Dynamic Individual IK) and return to use the IK position and rotation in the same animation.

- Now you can create your own State IKs instead of getting them automatically based on the Animator Controller.

IMPORTANT: Backup your project before updating to 1.1 version if you already have version 1.0. Old script configuration is not compatible with the new version. Previously positioned and rotated IK attachment Transforms will work but you will need to drag them in the new IK Helper Tool inspector.

Index

Overview

Thanks for using any of my assets.
IK Helper Tool is a script for helping you to fit my animations (or other animations) into your characters.

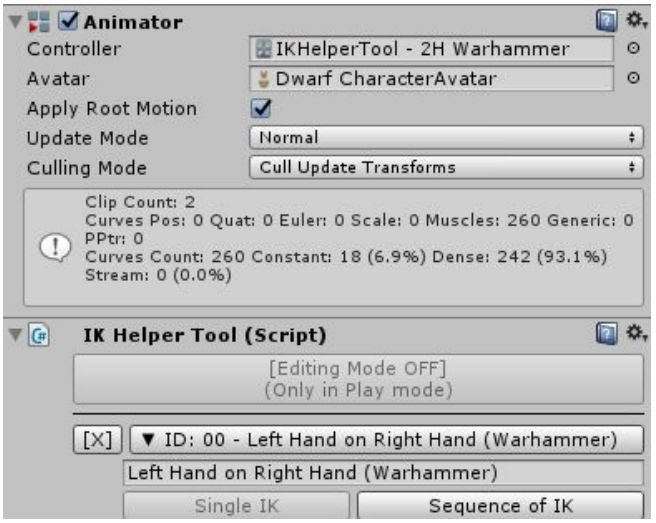
When retargeting using Unity Humanoid Mecanim system between two different skeletons with different length of arms or legs, the animation sometimes may not place the hand or foot in the correct place, especially when holding a prop with both hands. I've made IK Helper Tool to solve this problem or to help you to find the best workaround that fits your project best.

This documentation will give you a better understanding of how IK Helper Tool works.

Index

Main Principles

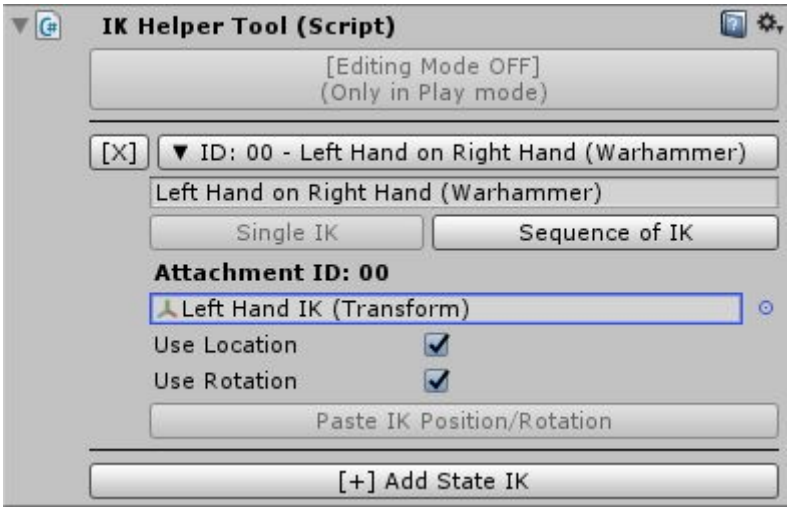
IK Helper Tool needs an **Animator** component to work.



The Animator Controller Layer must have IK Pass option enabled in order to make **IKHelperTool** work:

You will also need to create at least one empty GameObject for **IKHelperTool** to work. We will use the Transform of this new object as the IK position and rotation. We will call this **IK Attachment Transform**.

After creating the **IK Attachment Transform** we will need to configure a new **State IK** in the **IKHelperTool** script. We will drag this **IK Attachment Transform** to this new State. More information of how to set up a State here.



To make the IK finally work we will only need to call one of the starting IK functions (StartSingleIK or StartSequence). More information here.

You can call these functions from your custom scripts or use the ones I added that are included in the Animator Controllers inside the package.

These scripts are State Machine Behaviours (**IKHelperToolSMB**, **IKHelperToolSequenceSMB**, **IKHelperToolSMBRemover**). You can configure them by giving some variables like 'Delay' before IK to work and the 'Speed' to make them play smoothly.

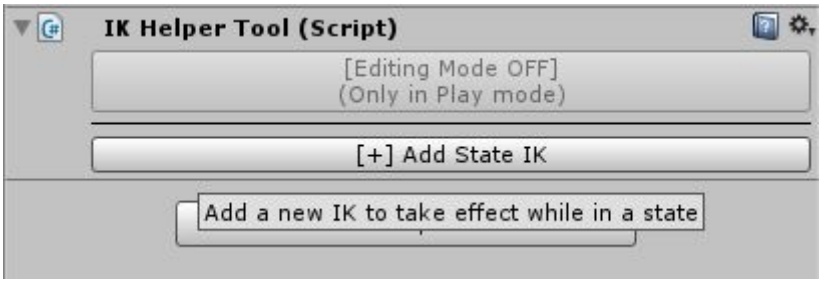
Index

How to Use

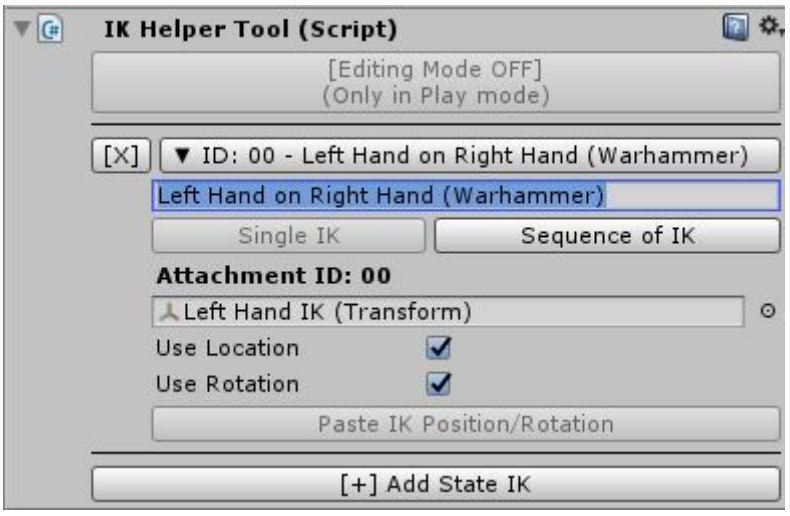
Main Script: IKHelperTool MonoBehaviour Script

This is the most important script. Needs to be added to your character in the same hierarchy level as the **Animator** component. After adding this component you will see a grey button for Editing Mode. Don't worry about it for the moment. The first thing you must do is to add a State IK with the '[+] Add State IK' button.

The **State IK** are groups of **IK Attachment Transforms** to be played when an animation is running. One animation can have multiple **State IKs** working at the same time but you will need one **State IK** for each different IK Goal you want to target (Right Hand, Left Hand, Right Foot or Left Foot).



You can assign a name for better identifying between multiple State IKs.



You will see two options: Single IK and Sequence of IK.

Single IK is the default and is used for simple animations where the IK stays still. Sequence IK is for using more than one **IK Attachment Transform** over time in the same animation. See the Mining example.

The **IK Attachment Transform** is a GameObject you will have to create and make it a child of a bone depending on the type of animation you want to fix. For a Sequence IK you will need more than one.

Fast examples of in which bone to set as child:

- Right Hand holds a weapon and looks fine. Left hand is holding the air. The **IK Attachment Transform** must be a child of the Right Hand. This will override the Left Hand animation holding the air and will follow the Right Hand.

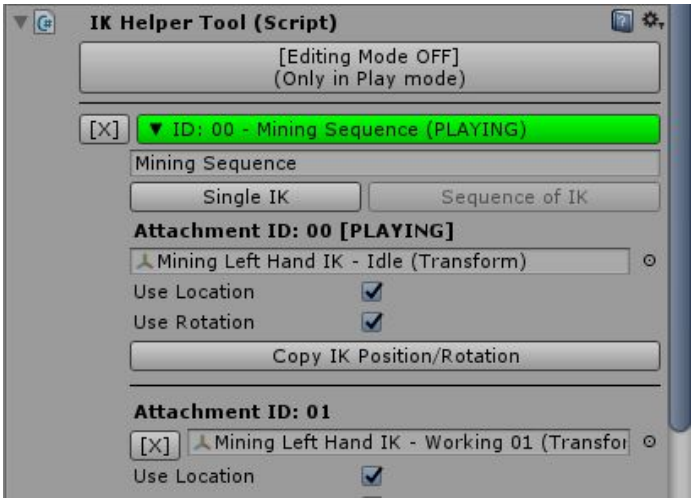
- Left Hand is passing through the floor. In this case the **IK Attachment Transform** is recommended to be a child of the skeleton root bone. We will use a Sequence IK and use the default animation except when the Left Hand reaches the floor.

For positioning and rotating the **IK Attachment Transform** in the desired values is recommended to use the Editing Mode (top button in the

custom inspector). Only available while in Play Mode. This mode freezes the animation and let's you freely find the correct position and rotation of **IK Attachment Transform**.

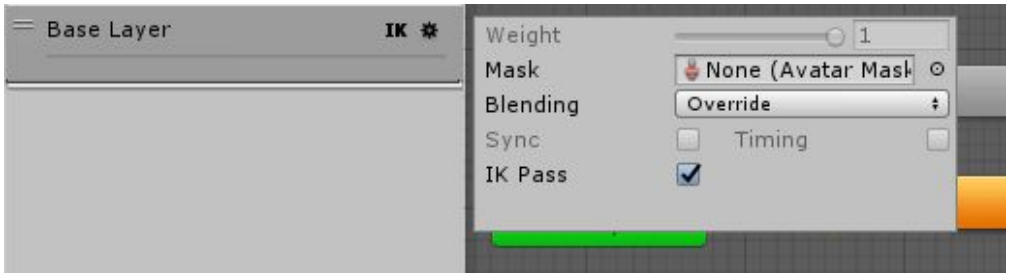
Don't forget to copy the values when still in Play Mode and paste them after exiting the Play Mode. You can also use the button 'Copy/Paste IK Position/Rotation' below each **IK Attachment Transform** entry in the custom inspector.

Make sure the **IK Attachment Transform** you are editing is actually working. You can check that in the custom inspector. Working IKs will appear with a [PLAYING] tag and the State IK will appear in green:



In order to make it actually playing you need to call the Single IK function or Sequence IK function from your custom script or from **IKHelperToolSMB** or **IKHelperToolSequenceSMB**.

Also make sure IK Pass is enabled in your Animator Controller Layer:



Functions from IKHelperTool that you can call:

```
public void StartSingleIK(int id, IKType goal, bool smooth, float speed)
```

id: Id number of the State IK to be activated.
goal: Right Hand, Left Hand, Right Foot or Left Foot.
smooth: If on will use speed parameter for smooth transition.
speed: Seconds that will the transition last (if smooth is enabled).

Description: Starts the first IK Transform Attachment in the State IK with the ID passed. Multiple IK Transform Attachment won't take effect, this is for Single IKs only.

```
public void StartSequence(int id, IKType goal, List<IKSequence> iKSequence, bool smoothEntry, bool clearOnExit)
```

id: Id number of the State IK to be activated.
goal: Right Hand, Left Hand, Right Foot or Left Foot.
smooth: If on will use speed parameter for smooth transition.
iKSequence: List of IKSequence with more parameters for each Transform Attachment (see IKHelperToolUtils.cs).
smoothEntry: If on will use speed parameter for smooth transition to the first IK Attachment.
clearOnExit: Clears last IK (sometimes useful if the next animation uses default animation).

Description: Starts the IK Transform Attachment sequence in the State IK with the ID passed. This function supports multiple IK Transform Attachments (Sequence IKs).

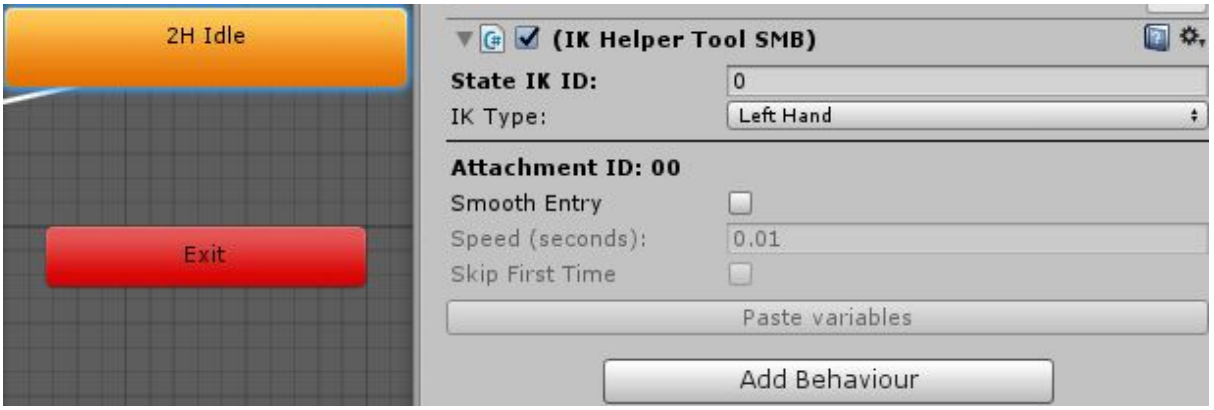
```
public void ClearIK(bool clearAll, int id, float delay, bool smooth, float speed)
```

`clearAll`: Clears all active IKs from all States.
`id`: Individual ID of the State to be cleared (if ClearAll is off).
`delay`: Seconds to wait before clearing the IK.
`smooth`: If on will use speed parameter for smooth transition.
`speed`: Seconds that will the transition last (if smooth is enabled).

Description: Stops playing one of the States IK or all of them.

State Machine Behaviour Scripts

You can add these scripts as State Machine Behaviours in the animation states of your Animator Controller:



You will need to add at least one of these scripts in order for the IK actually work or use your own custom script. Your custom script doesn't need to be a State Machine Behaviour but must call either StartSequence or StartSingleIK properly.

IKHelperToolSMB StateMachineBehaviour Script

This script calls StartSingleIK as soon as the animation on the animator state starts. Allows you to configure some of the parameters for the StartSingleIK function.

IKHelperToolSequenceSMB StateMachineBehaviour Script

Same as the previous script but the Sequence version. This script calls StartSequence as soon as the animation on the animator state starts. Allows you to configure some of the parameters for the StartSequence function.

IKHelperToolSMBRemover StateMachineBehaviour Script

Removes one or all of the active IKs as soon as the animation on the animator state starts. You can configure some delay and speed.

One known issue regarding State Machine Behaviours, when the animation is a loop and the animator state doesn't change the IK may not work. To fix this you can add a transition to the same state like in this image:

Index

Examples

1. **One hand is holding a two-handed prop and the other hand is misplaced holding the air (Single). Difficulty: Easy.**
   See Human or Dwarf 2H Sword prefabs in the example scene. In this case the IK must be a child of the opposite hand which holds properly the prop.

2. **One hand is passing through the floor or a wall (Sequence). Difficulty: Moderate.**
   See Dwarf Shockwave prefabs in the example scene. In this case the IK is a child of the root skeleton bone.

3. **One hand is holding a two-handed prop and the other hand is misplaced holding the air and it moves during the animation (Sequence). Difficulty: Moderate.**
   See Dwarf Mining prefabs in the example scene. In this case the IKs used in the sequence must be children of the opposite hand which holds properly the prop.

4. **Both hands are supposed to hold a prop but none of them fit properly (Single for both hands). Difficulty: Hard.**
   See Dwarf with Rifle prefabs in the example scene. In this case we are gonna take the hand that holds better the prop. The IK Transform of this hand will need to be a child of the hand parent and will need to have the IK Transform of the other hand as a child.

Index

Contact Support

For support, questions or suggestions regarding this product send me an email to:
support@keviniglesias.com

Kevin Iglesias - www.keviniglesias.com

- **More Unity Assets**
- **Facebook**
- **Twitter**
- **YouTube Channel**