

# Throw Move Creation Guide

This page will guide you through how to create a throw move. We will reference the [Active Frames](#) and [Opponent Override](#), so it's recommended you keep these pages opened for handy reference.

Also worth looking at is Robot Kyle's throw moves to get a handle on what's required. They are located in:

```
.\\UFE\\Characters\\Robot Kyle\\Moves\\ThrowAttempt.asset  
.\\UFE\\Characters\\Robot Kyle\\Moves\\ThrowConfirm.asset  
.\\UFE\\Characters\\Robot Kyle\\Moves\\ThrowReaction.asset
```

## Prerequisite:

You will require 3 animations. One for each of the above moves. Ensure the Throw Confirm and Throw Reaction clips are synced.

## VERY IMPORTANT

Your characters **must** have a Throw Collider hitbox assigned similar to image below. The *bodypart*, *link*, *shape* and *dimensions* will depend on your own requirements, but it must have Collision Type of **Throw Collider**. In the scene view, it's the red hitbox.





## Attempt, Confirm and Reaction

There are at least 3 moves required for a throw. The **Throw Attempt**, the **Throw Confirm** and the **Throw Reaction**. An optional move is the **Throw Tech** if you want those included in your game.

The Throw Reaction is the opponent's reaction to the throw. The Throw Attempt is the move with throw's **Input** and is what the player executes. The Throw Confirm is the move that is cast once the Throw Attempt is considered successful.

The reason we have a Throw Attempt move *and* a Throw Confirm move is to allow for whiffed/missed throw attempts. If the opponent was out of range during the Throw Attempt, then it will not confirm into the Throw Confirm move and continue with the Throw Attempt animation. Create your Throw Attempt animation with a decent length recovery - this will give you a long whiff animation for when the attempt misses.

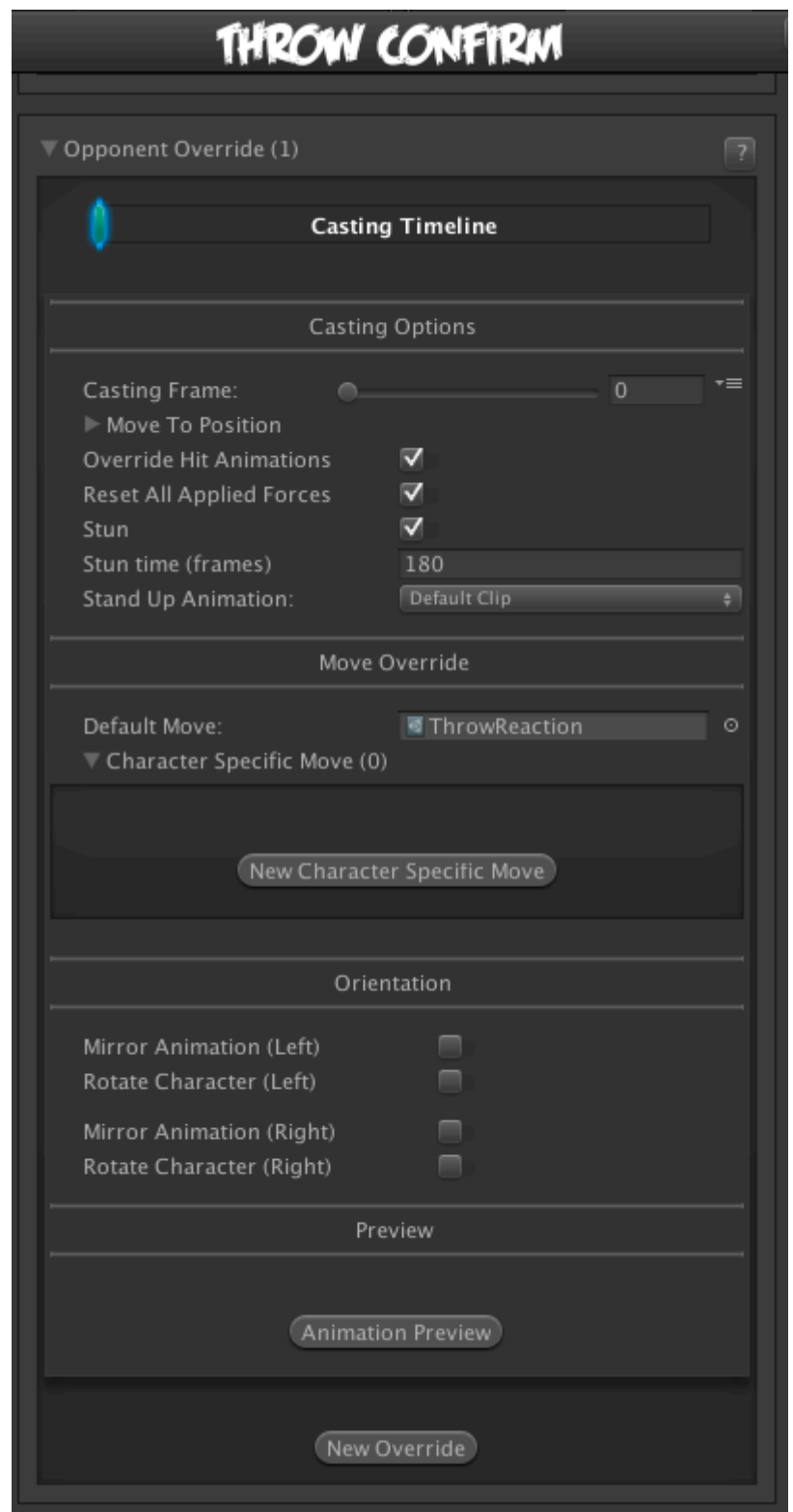
---

## Throw Attempt



1. Out of the 3, this is the only move that has an [Input](#).
2. In [Active Frames](#), set the Hit Confirm Type to be Throw.
3. In Hit Conditions, set the state you want this throw to be successful. If the throw can be combed/linked, you'll want Stunned toggled on. If you want OTG throw, you'll want Down on.
4. When created, drag the Throw Confirm move into the Throw Move Confirm.
5. If the throw is techable, toggle that on too and drag the Throw Tech move in.

## Throw Confirm



1. This move should have no input.
2. In [Opponent Override](#), ensure the Default Move has the Throw Reaction move assigned. If you have character specific reactions, also add them here too.
3. Set Move To Position if required.
4. Set Stun on to prevent opponent from reacting during the override. Match the Stun time with at least the same length as the throw's animation plus the knockdown time.
5. Override the Hit Animations if you have reactions already in the Throw Reaction move.
6. In [Active Frames](#), this should now be set like a normal Hit type attack (not Throw type, we already set that for the Throw Attempt). Match the hurt boxes and frame data with the hit

impacts of the animation.

7. If you set Override Hit Animations in Opponent Override, then the Hit Strength only affects the Hit Effect and not the opponent's reaction animation.

---

## Throw Reaction

1. This is straight forward. It's just a move with an animation. It has no Input, no override, no conditions and no active frames.

---

**IMPORTANT NOTE:** For the throw to work against all characters, every character needs to have the Throw Reaction move in their move set.

If characters are using unique rigs or different animation types, then you'll need to create a reaction move for specific rigs or animation types. Ensure [Character Specific Reaction](#) is set with the appropriate reaction move for the character.

---

[<- Back to UFE Home Page](#)

From:

<http://www.ufe3d.com/> - **Universal Fighting Engine**

Permanent link:

[http://www.ufe3d.com/doku.php/throwtutorial:throw\\_tutorial?rev=1414534410](http://www.ufe3d.com/doku.php/throwtutorial:throw_tutorial?rev=1414534410)

Last update: **2014/10/28 18:13**

