

Table of Contents

About this primer.....	1
ComboRules Table.....	2
Animation Montage (Combo Montage).....	4

About this primer

The combo system in Conan Exiles works with two fundamental ways of controlling it; the ComboRules table, and the Animation Montages that contain (amongst other things) the animation for the attack. In this primer, we will go through both of these aspects, as well as touch on some other, important, information about the combat in Conan Exiles.

ComboRules Table

The Combo Rules table can be found here:

`/Game/Systems/Combat/ComboSystem/DT_ComboRules`

Below, we will go through the table in detail.

WeaponComboTypeLead - this dropdown is populated from the "E_WeaponComboType" enum. In the Item-table, the Visual Object for weapons contains a blueprint (for example "claws2h_werewolf_player") - and in that blueprint, the ComboType of the weapon needs to be set to this in order for the game to know what combos the player is executing. Changing the selection in the Visual Object would make the weapon use a completely different combo-rules set.

WeaponComboTypeSupport - this dropdown determines the equipped off-hand weapon. For a two-handed weapon, leave this blank. For a one-handed weapon, leaving this blank determines what combo-rules are executed when only the Lead type weapon is equipped. If you want separate combos for each type of Supporting (off-hand) weapon, you will need to make combo-rules for each combination of off-hand weapon.

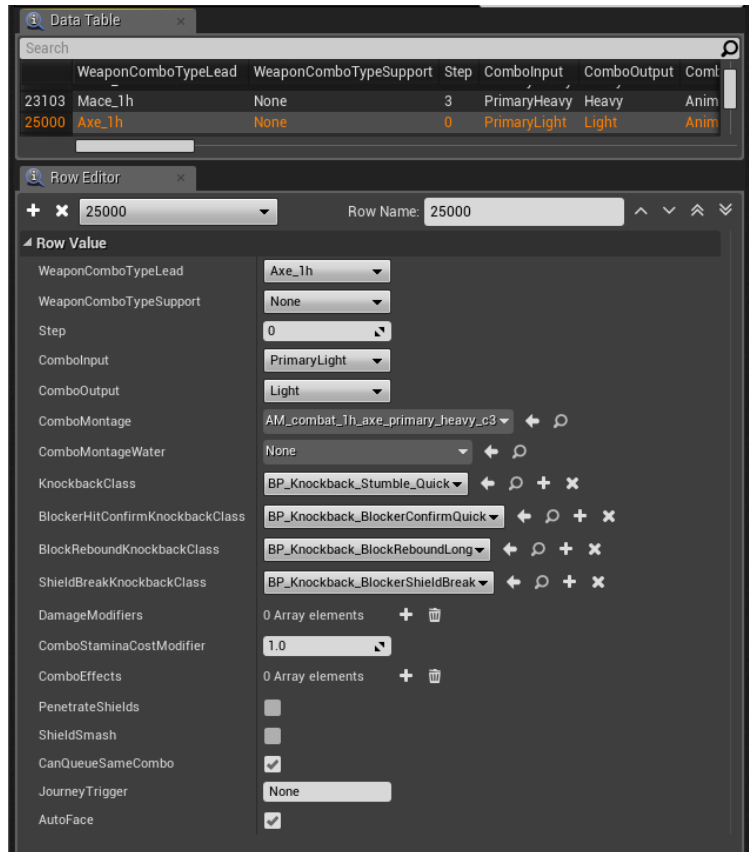
Step - For a new weapon, this must start at 0. This is basically the sequence value of the attack in a given combo. It starts at 0 and can go up to however much you want. By default, the Conan Exiles weapons usually only go up to Step 3 (although there are exceptions). It's important to know how the Animation Montages tie into this (read more about this later).

ComboInput - This is either PrimaryLight, PrimaryHeavy, or Off-Hand (essentially - what would the player press to execute this attack)

ComboOutput - This is used for other systems to distinguish the type of attack between light or heavy. For example, there are certain perks that affects only light or heavy attacks.

ComboMontage - This is the animation montage of the combo, and is handled separately below. This is a very important step to set up correctly.

ComboMontageWater - This is the animation montage when the weapon is used under water.



KnockbackClass, BlockerHitConfirmKnockbackClass, BlockReboundKnockbackClass, ShieldBreakKnockbackClass - The knockback class determines what type of knockback (if any) the attack has under certain conditions. These are all blueprints stored in the "" folder. Each of these blueprints allow you to set up unique knockback classes if you want to - allowing weapons to (quite literally) fling enemies around as if they were paper-dolls.

DamageModifiers - The entries in this list gives you control over additional modifiers to damage on the target - each type of damage (typically Health, Concussive, Harvest) can have separate modifiers here. The modifiers here are multiplicative - a heavy Mace finisher move does much more damage than the first attack (step 0) in the sequence.

ComboStaminaCostModifier - This multiplies the amount of stamina the attack costs to perform.

ComboEffects - Here you can control (De)buffs applied on the target when hit by this step in the combo. These blueprints are all children of the "Damage Type" class, and in the blueprint, the "Effect type" directly links to a buff (to read more about buffs, please refer to the "**Creating Buffs**" document found in this .zip file). You can also start these buffs with a predesignated Potency.

PenetrateShields - If this is ticked, the attack ignores shields

ShieldSmash - If this is ticked, there is no bounceback from hitting shields, blocking counts, and attacker plays the block-response (BlockHitConfirm)

CanQueueSameCombo - If this is ticked, it's possible to queue the same combo. This is always checked unless there are specific reasons why you don't want to allow the same type of combo to be repeated. An example of this would be the werewolf boss that cannot spam the same combo over and over.

JourneyTrigger - Certain attacks (finishers, attacks with bleeds, etc) can trigger certain Journeysteps to complete.

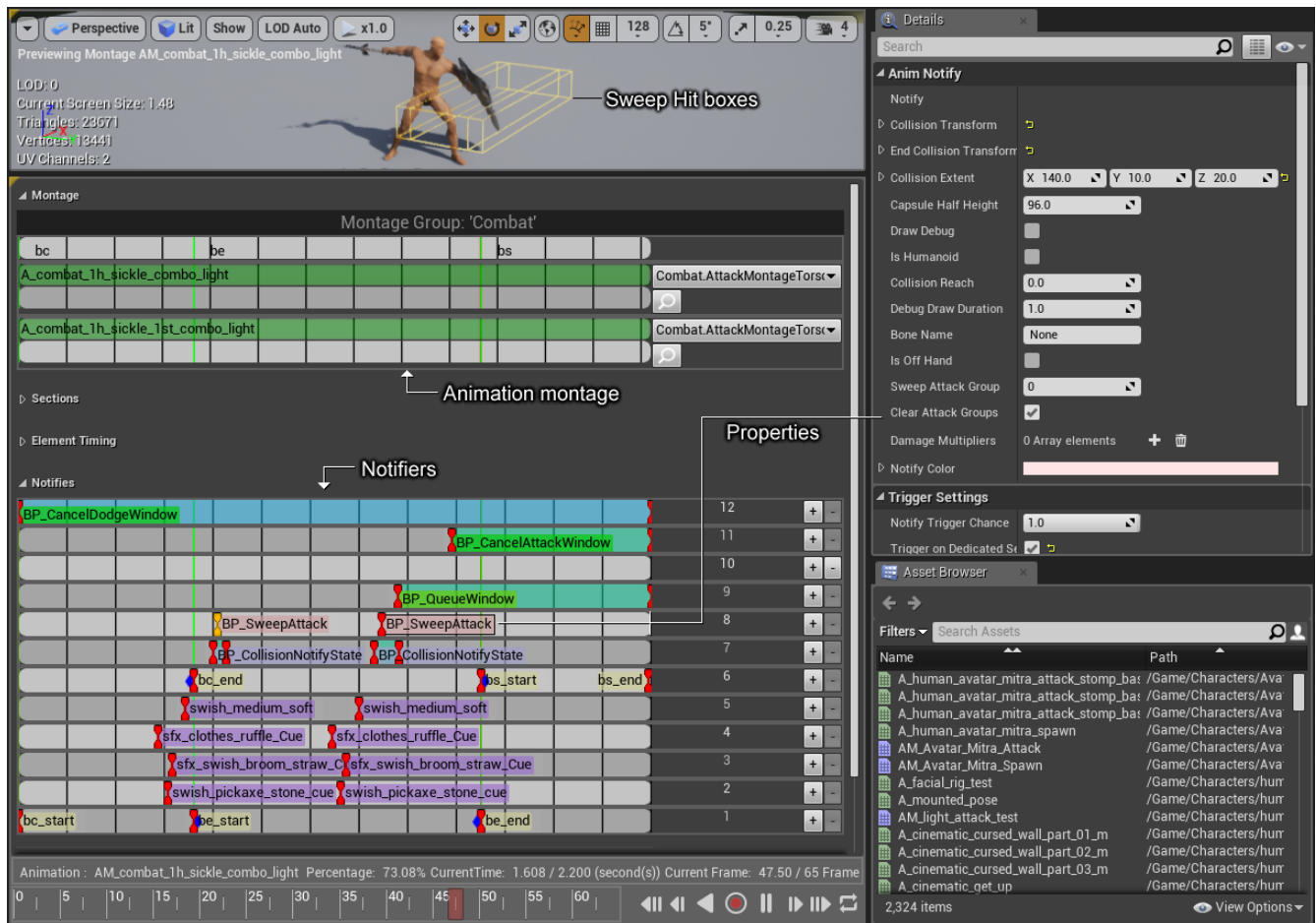
Autoface - Typically clicked, except for off-hand shield blocking

Animation Montage (Combo Montage)

While the ComboRules Table determines the RPG stats of the weapon attacks during the attacks in a combo and governs what attacks can be done in which order, the Combo Montages determine all the nitty-gritty details for each animation, such as HyperArmor, Cancel windows, recovery and so forth.

In this section, we will go through the Animation Montage bit by bit in an effort to get to grips with what does what in the animation montage. If you haven't touched animation montages at all, it might be a good idea to read up on the basics for them here:

<https://docs.unrealengine.com/en-us/Engine/Animation/AnimMontage>



As far as the comborules go, the Notifiers are the most interesting properties here, although it's worth mentioning that the SweepAttack box is the actual hitbox for the attack. That said, let's go into what the different notifiers that pertain to combo-rules do.

BP_CancelDodgeWindow

During this notifier, it's possible to use a dodge move to cancel the rest of the animation

BP_CancelAttackWindow

If the player takes damage during this duration, the animation is cancelled. If hyperarmor is present in the attack, this should not overlap the hyper-armor attack duration.

BP_SweepAttack

This is the hit collision for when NPCs use the attack

BP_CollisionNotifyState

This is the hit collision for when players use the attack

BP_HyperArmorWindow

During this notifier, the player performing the animation has hyper-armor, meaning that the attack cannot be cancelled.

BP_CancelMoveWindow

During this notifier, the player can begin moving to cancel the rest of the attack.

BP_ComboWindow

During this window, if a player makes an attack, the system attempts to set the next attack to be the proper Step to execute through the ComboRules Table. If the Comborules table cannot find an entry, it resets it to Step 0. This an important feature - without this, the system has no way of knowing when the player can queue up the next attack.

BP_CapsuleSlidewindow

Stops playesr from moving into other players/npcs during this attack (checks the attackers capsule vs. the defenders mesh).

BP_QueueWindow

At first glance you may think this does the same thing as the ComboWindow, but there are important differences. The Combo-Window attempts to increase the step of the next attack, while the QueueWindow is there to start the next attack from Step 0.

BP_DisableCharacterRotation

During this notifier, it's not possible to rotate the player (may not work).

BP_WeaponTrailNotifyState_xx (Cripple)

This notifier controls the type of weapon trail should be present in the attack (Cripple, Sunder, Bleed, etc).

BP_RotationRateModificationNotifyState

This multiplies the rotation rate for NPCs during the attack. This allows you to set up custom rotation rates during parts of the animation (such as locking down rotation completely, or multiplying it to grant NPCs a higher chance to rotate towards the player in time).

A note about sweep boxes

Sweep boxes can be overridden (and in many cases they are) in the property window. When such overrides are not in place, it uses the default collision of the weapon-mesh, which may not be desirable. It's also possible to open the weapon blueprint (for example, "Axe1h") and set up custom weapon-"parts" (a function called "GetSpecificCollision").