TotK Damage Calculations

Credits/Resources

Written by : echocolat
First experiments and prototype formulae by : echocolat & crimsonstarfall
Deeper experiments by : dt12345, KreaTV1, phil_macrocheira, echocolat
Data from the game files, also can be found here (thanks phil_macrocheira)
Quick calculator here (make a copy to use, File -> Create a copy)
Old, kinda outdated Reddit post explaining the same things here

Introduction

Tears of the Kingdom combat's system is quite deep and complex. In this document we will only cover the damage the Player deals to other entities. Its core is still Breath of the Wild's, but there are additions and tweaks that we also need to talk about. Let's go!

The calculations

A basic formula

We'll start with the general formula for the damage the player deals with a melee weapon, I'll explain the terms a bit later don't worry.

$$FinalDamage = floor(TrueDamage * multipliers)$$

TrueDamage

$floor$ is the mathematical function that rounds down a number. Let's talk a bit about $TrueDamage$. It's basically the "normal" damage of the weapon, counting fuse damage, attack up modifiers and the base attack of the weapon. So theoretically, it would just be:

$$TrueDamage = BaseAttack + AttackUpMod + FuseDamage + ZonaiDamage$$

Let's talk a bit about all those terms. $BaseAttack$ is the base damage of the weapon. Keep in mind I'm talking about the $BaseAttack$ parameter as stated by the game files. $AttackUpMod$ is the value of the Attack Up / Attack Up+ modifier, if it exists on the weapon. Attack Up always ranges between 3 and 5, and Attack Up+ always ranges between 6 and 10. $FuseDamage$ is the base fuse ($AdditionalDamage$) damage of whatever is fused to the weapon, if the weapon has been fused, multiplied by the $AttachMulAttackValue$ of the weapon, which is 2.0 if the weapon is a gerudo weapon, and 1.0 otherwise. Keep in mind I'm

talking about the $AdditionalDamage$ parameter as stated by the game files. Finally, $ZonaiDamage$ is the damage coming from the Zonaite-Powered weapons if fused to a Zonai-based thing. It's always the value of $AttachZonauAttackValue$ which is set to 3 if the weapon is a tier 1 Zonaite weapon, 5 if the weapon is a tier 2 Zonaite weapon, 10 if the weapon is a tier 3 Zonaite weapon.

There's a catch

This is where things start to be complicated. For some reason, Nintendo developers decided to make Spear-type weapons display more damage than what they actually do, and to make Two Handed-type weapons display less damage than what they actually do. So, how does that work? Deep within the game's code, lie four numbers that we decided to call the weapon type base multipliers. Two of them, called *Two handed multiplier* and *Spear multiplier*, affect the damage that the game displays in the Hyrule Compendium and in the Inventory. And the other two, called *Two handed anti-multiplier* and *Spear anti-multiplier* (the name comes from the fact that they are the inverse of the base multipliers) affect the true amount of damage the weapon deals. Let's go over them, one by one.

Two handed weapons

The *Two handed multiplier* is equal to 0.95. It means the Two handed-type weapons display about 95% of their damage in the Inventory and in the Hyrule Compendium. The true formula is:

$$ShownDamage = floor(BaseAttack * 0.95)$$

The *Two handed anti-multiplier* is equal to 1.052632. It multiplies every term in $TrueDamage$ except for $BaseAttack$. Which leaves us with the following formulae for $TrueDamage$ and $ShownDamage$ for the Two handed-type weapons.

$$TrueDamage = BaseAttack + floor((AddedDamage) * 1.052632)$$
$$ShownDamage = floor(BaseAttack * 0.95) + AddedDamage$$

Where $AddedDamage = AttackUpMod + FuseDamage + ZonaiDamage$.

You're probably lost at this point so let's do a quick example. Let's take a Gerudo Claymore✨ with Attack Up+ set to 10, fused to a Silver Bokoblin Horn. The Gerudo Claymore✨ has a $BaseAttack$ of 24, a $AttachMulAttackValue$ of 2.0 (because Gerudo weapon), no $ZonaiDamage$, and Silver Bokoblin Horn has a $AdditionalDamage$ of 31. When we put all of this in the formula, we get:

$$TrueDamage = 24 + floor((10 + 2 * 31 + 0) * 1.052632) = 99$$
$$ShownDamage = floor(24 * 0.95) + (10 + 2 * 31 + 0) = 94$$

I'll make another example with Zonaite-Powered weapons. Let's take a Mighty Zonaite Longsword with Attack Up+ set to 10, fused to a Flux Construct III Core. The Mighty Zonaite Longsword has a $BaseAttack$ of 16, a $AttachMulAttackValue$ of 1.0 (because no Gerudo

weapon), a $ZonaiDamage$ of 10, and Flux Construct III Core has a $AdditionalDamage$ of 32. When we put all of this in the formula, we get:

$$TrueDamage = 16 + floor((10 + 1 * 32 + 10) * 1.052632) = 70$$
$$ShownDamage = floor(16 * 0.95) + (10 + 1 * 32 + 10) = 67$$

Spears

The *Spear multiplier* is equal to 1.326856. It means the Spear-type weapons display about 133% of their damage in the Inventory and in the Hyrule Compendium. The true formula is ($ceil$ is the rounding up function):

$$ShownDamage = ceil(BaseAttack * 1.326856)$$

The *Spear anti-multiplier* is equal to 0.7536613. It multiplies every term in $TrueDamage$ except for $BaseAttack$. Which leaves us with the following formulae for $TrueDamage$ and $ShownDamage$ for the Spear-type weapons.

$$TrueDamage = BaseAttack + ceil((AddedDamage) * 0.7536613)$$
$$ShownDamage = ceil(BaseAttack * 1.326856) + AddedDamage$$

Where $AddedDamage = AttackUpMod + FuseDamage + ZonaiDamage$.

Let's do two more examples to make sure you understand. Let's take a Royal Guard's Spear (decayed), with Attack Up+ set to 10, fused to a Silver Lynel Saber Horn. The Royal Guard's Spear has a $BaseAttack$ of 11, a $AttachMulAttackValue$ of 1.0 (because not a Gerudo weapon), no $ZonaiDamage$, and Silver Lynel Saber Horn has a $AdditionalDamage$ of 55. When we put all of this in the formula, we get:

$$TrueDamage = 11 + ceil((10 + 1 * 55 + 0) * 0.7536613) = 60$$
$$ShownDamage = ceil(11 * 1.326856) + 10 + 1 * 55 + 0 = 80$$

I'll make a last example with Zonaite-Powered weapons. Let's take a Strong Zonaite Spear with Attack Up+ set to 10, fused to a Captain Construct IV Horn. The Strong Zonaite Spear has a $BaseAttack$ of 16, a $AttachMulAttackValue$ of 1.0 (because not a Gerudo weapon), a $ZonaiDamage$ of 5, and Captain Construct IV has a $AdditionalDamage$ of 35. When we put all of this in the formula, we get:

$$TrueDamage = 4 + ceil((10 + 1 * 35 + 5) * 0.7536613) = 42$$
$$ShownDamage = ceil(4 * 1.326856) + 10 + 1 * 35 + 5 = 56$$

multipliers

Now that we have the $TrueDamage$, let's check all the multipliers that… multiply it. I'll make a list of them all, explaining each time how they work and their value.

- $DamageRate$ is basically the “kind” of damage that your weapon deals. For us, there are only three kinds of “kind of damage” we need to worry about, which are $LargeWeapon$ (Two handed-type weapons), $SmallWeapon$ (One handed-type weapons) and $ThrustWeapon$ (Spear-type weapons). Each rate is defined per damageable entity, you can check all of them here.
- The $Attack\ Up$ multiplier is set to 1.0 when the player has not any Attack Up effect from meals and/or armors, 1.2 when the player has Attack Up level 1, 1.3 when the player has Attack Up level 2 and 1.5 when the player has Attack Up level 3. It caps at level 3.
- The $Bone\ Proficiency$ multiplier is set to 1.8 when either the base Weapon or the fuse material has the $Bone$ property. If both have the $Bone$ property, the multiplier is only applied once.
- The $Critical\ Hit$ multiplier is set to 2.0 when the Player deals the last hit of their standard combo.
- The $Thrown\ Weapon$ multiplier is set to 2.0 if the weapon has been thrown by the player if it’s not a boomerang. A boomerang’s $Thrown\ Weapon$ multiplier is set to 1.5.
- Hitting a frozen enemy triggers the $Ice\ Shattering$ multiplier which is in most cases 3.0, but is entity-dependent (see $FrozenBreakDamageRate$ column in this spreadsheet).
- If and only if the weapon has not been thrown, is not dealing a critical hit and/or shatters a frozen enemy, a weapon at 1 durability will trigger the $One\ Durability$ multiplier, set to 2.0.
- Dealing a hit to the entity’s weakpoint (if it exists) will trigger the $Weakpoint$ multiplier, that is entity-dependent (most likely 2.0).
- Doing a streakstrike triggers the $Streakstrike$ multiplier set to 8.0.
- Dealing a shattering hit on a rock-type entity will trigger the $Rock\ shattering$ multiplier, set to 1.5 for Two handed-type weapons, and 1.25 for One handed-type and spear-type weapons.
- All the Weapon traits multipliers are set to 2.0 if their requirement is met:
    - $Breaking\ Point$ multiplier applies if the weapon is wielded by Link and has 3 durability or less (shown by the UI).
    - $Desperate\ Strength$ multiplier applies if the weapon is wielded by Link and if Link has one heart or less left (shown by the UI).
    - $Improved\ Flurry\ Rush$ multiplier applies if the weapon is wielded by Link and if Link is doing a flurry rush (not shown by the UI).
    - $Improved\ Sneakstrike$ multiplier applies during a sneakstrike (on top of the $Sneakstrike$ multiplier).
    - $Water\ Warrior$ multiplier applies if the weapon is wielded by Link and is wet.

To expand further…

Skip this section if you care only about melee damage, since I said all I had to say for melee damage. This section aims to generalize the damage formula to all cases (arrows, shields, enemies, animals).

Let's call our previous $TrueDamage$ "$BaseDamage$" now, when we're considering weapon damage. Otherwise, $BaseDamage$ will just be the base damage of the entity dealing the damage. The basic formula we had earlier will transform to:

$$FinalDamage = floor(TrueDamage * mults) + other$$

So, what is $TrueDamage$ now? I'll give you a new formula for it, and I'll discuss the new terms inside it.

$$TrueDamage = EntityPower + BaseDamage - Armor$$

$EntityPower$ is an entity-dependent quantity that is either the $BaseAttackPower$ parameter if the entity is not wielding any kind of weapon, $BaseAttackPowerMeleeWeapon$ parameter if the entity is dealing the damage with a melee weapon, or $BaseAttackPowerRangedWeapon$ parameter if the entity is dealing the damage with a ranged weapon. You can check them all here.

If Link is the entity receiving the damage, the $Armor$ term is used. That term can be calculated quite easily with the following formula:

$$Armor = BaseArmor + MealArmor$$

Where $BaseArmor$ is the combined armor value from all armor pieces the Player is wielding, and $MealArmor$ is 4 if the Player has a Defense Up meal effect level 1, 12 if it's level 2, and 24 if it's level 3.

The $multipliers$ term also gets new multipliers included in, as well as a slight tweak for Bone Proficiency:

- Bone Proficiency can apply on arrows if they are shooted by a $Bone$ bow. However, it DOES NOT apply if the fuse material attached to the arrow is $Bone$.
- The $RodLv$ multiplier applies if the damage source comes from magic wands. It's equal to 1.0 except for Magic Rod, Magic Scepter and Magic Staff, for which it's equal to 2.0.
- There is a set of three "elemental" weaknesses that apply if the target is not straight up one shot because they have the one shot death procedures:
    - The $FireOnWater$ multiplier is equal to 1.5 and is used when the attacker entity is Fire-type and the receiver is Water-type.
    - The $FireOnIce$ multiplier is equal to 2.0 and is used when the attacker entity is Fire-type and the receiver is Ice-type.
    - The $IceOnFire$ multiplier is equal to 2.0 and is used when the attacker entity is Ice-type and the receiver is Fire-type.

Finally, let's discuss $other$. This is basically any additive damage that has not been discussed yet.

- If the attacker entity is either Riju's lightning or Weather lightning, the $Lightning$ additive damage is added, if the target isn't immune to lightning (see ConditionParam

in the same spreadsheet). It is entity-dependent (see $LightningDamageParameters$ here).

- If the attacker entity is elemental, either fire, ice, water or electric, the $Elemental$ additive damage is added, if the target isn't immune to said element. It also is entity-dependent, see $[element]DamageParameters$ in the same spreadsheet as above.
- Finally, if the attacker entity is a strong light entity, the $Strong\ Light$ additive damage is added, if the target isn't immune to it. It is also entity-depenedent, see $LightShockDamageParameters$ in the same spreadsheet as above.

Conclusion

That's all for melee damage ! I hope it was crystal clear. If you still have questions for me, please ask me privately on Discord, my @ is echocolat there. I'd like to thank again all my friends and colleagues dataminers in the Datamining server, especially dt12345, KreaTV1, Doge228 and phil_macrocheira.