TL; DR

- Moves have Categories, which can be seen in Table 1.
- Similarly, Moves can make contact (think Spiky Shield), which can be seen in Table 2.

Contents

0.1 What Is `MoveData`?

`MoveData` is an asset that contains static data on a Move. Specifically, it contains information on:

- **Base power**. A measure of the relative power of a Move. Starter Moves may only have 10–30 base power, whereas the most powerful moves may tip the scales at 100. The majority of Moves should have low base powers—we don't want the OHKOs of Pokemon, as this isn't a turn-based game.
- **Base cooldown**. The cooldown in seconds. This gets modified by the `CombatStatsComponent`'s Haste.
- **Category**. A representation of the kind of Move this is. The `MoveCategory`s are:

Table 1: All `MoveCategory`s currently implemented.

Name	Description
`None`	Default. Don't use this. If you see it, it's an error.
`PhysicalDamage`	If damaging, the Move relies on the user's physical abilities, like strength.
`SpecialDamage`	If damaging, the Move relies on the user's non-physical abilities, like psychic or spiritual ability.
`PhysicalHealing`	A move that restores health (self, allies, enemies, etc.) by physical means. For example, an injection, food, a bandage, or acupuncture.
`SpecialHealing`	A move that restores health (self, allies, enemies, etc.) non-physically. For example, mental healing, emotional support, spiritual peace, or the reversing of time to heal wounds.
`Summoning`	A move that summons an object or creature.
`Utility`	Does no damage to opponent or self. May still attach effects. Pokemon example: Thunder Wave or Pay Day (but without damage).

- **Contact**. Determines what kind of contact (if any) occurs when the Move hits a target. The `MoveCategory`s are:

Table 2: All `MoveContact`s currently implemented.

Name	Description
`None`	No contact (e.g., Summoning or damaging from afar).
`PhysicalContact`	The damage or status requires the user to make physical contact with the target. Example: punching something.
`SpecialContact`	The damage or status requires the user to make non-physical contact with the target. Example: direct psychic connection.

  I'll concede that `SpecialContact` probably isn't super useful to track, but you never know. It's easier to put it in now rather than later.

- **Display name**. If the field `DisplayName` is blank, the asset's name will be used. May be useful for special characters.

- **Effects to inflict**. A `TMap` of `EffectComponent`s and thier percentages of being applied on a successful Move hit. Pokemon example: a Move having 30% chance to Burn would have its `TPair` be `Burn` and `30`.

- **Randomness**. A range of values to multiply to calculate the final damage. Normally, any Move's damage is multiplied by a number between 0.85 and 1 (range taken, of course, from Pokemon). This variance makes it a little more exciting.

- **Supporting text**. `SupportingText` includes dev notes that doesn't appear in-game, a proper description, and flavor text (e.g., for hovers).

- **Move type(s)**. The incredibly vast majority of moves should only associate with one Type. It gets far too complicated and unpredictably unfun to have multi-type attacks.

0.2 Damage Calculation

The formulas for damage and healing are inspired by Pokemon. Like it or not, that's the gold standard and what most players will expect when judging base powers, stat values, etc.

The formulas are implemented in `CombatStatsComponent.cpp`:

$$\text{Damage} = \left[(0.4 \times \text{Level} + 2) \times \frac{\text{Base Power}}{50} \times \frac{\text{Attack}}{\text{Defense}} + 2\right] \times \text{Crit Multiplier} \times \text{Random Fluctuation} \times \text{STAB Multiplier} \times \text{Type Advantage}$$

where:

- Level is the attacker's Level

- Base Power is determined by the `MoveData`

- Attack and Defense are the appropriate stat values

- Crit Multiplier is 1.5× if a critical hit occurred; 1.0× otherwise
  - As outlined in the Critical Hit section of the Stats documentation, if a crit does occur, this reverts Defense to its un-buffed value (including everything, e.g., Mutations).
  - If the Critical Hit stat is above 100%, the critical damage increases accordingly. For example, if Crit is 120%, the damage bonus is $1.5 + 0.2 = 1.7\times$.
  - This can be a dangerous game mechanic and will require a lot of balancing. However, it also opens the strategies in an otherwise traditional stat scheme.
- Random Fluctuation is determined by the "randomness" of the `MoveData` (in particular, the field `MoveData.RandomRange`). Nominally, this is 0.85–1.
- Same Type Attack Bonus (STAB) rewards Monsters for using Moves that share their Types (e.g., a Fire-type Monster using a Fire-type Move). Nominally, this is 1.5×, but can be more or less (e.g., for Typeless Monsters, it's 2×).
- Type Advantage is outlined in the Types documentation. Normally, it ranges anywhere from −1 (Electric actually heals Electric) to 0.25× (double resisted) to 4× (doubly weak). It follows established Pokemon conventions.

Similarly, the healing formula is:

$$\text{Healing} = \left[(0.4 \times \text{Level} + 2) \times \frac{\text{Base Power}}{50} \times \frac{\text{Attack}}{100} + 2\right] \times \text{Crit Multiplier} \times \text{Random Fluctuation} \times \text{STAB Multiplier}$$

Clearly, the Defense of the healee should not inhibit the heal, so a static 100 has been used instead (as 100 is an average stat). Moreover, Type advantages should not play a role in healing. Heals can crit, though.