

Divining the Combat System in the Settlers Online

Foreword. This paper is a short summary of how the combat mechanics work in the Settlers Online—based on scattered and scarce pieces of information from the forums from the developers, as well as testing on the main and pre-staging servers. Whenever a certain piece of information is not verified, it will be explicitly noted.

As an aside, the authors consider the adventure combat model only, and discard the expedition combat model as trivial and uninteresting.

1 Units

Each unit has a *type*, prescribing the base attributes governing their behavior in a battle. These are:

- hit points, which we will denote with H_e ;
- experience, Exp —the amount of experience points the attacker gets when this unit is killed;
- low and high damage, denoted \underline{Dm} and \overline{Dm} , respectively;
- accuracy, Acc , which is the probability of dealing high damage rather than low damage;
- splash chance, Spl —probability that the inflicted damage will spill over to the next defender.

All of them, except for Acc and Spl are integer-valued. Furthermore, each of the attributes can have *modifiers*. Modifiers come in two flavors: absolute value modifier (additive modifier), and relative/rate modifier (multiplicative modifier). If A is the base value of any attribute, and (x, r) is the pair additive modifier/multiplicative modifier, then

$$A + \llbracket r A \rrbracket + x$$

will be the effective value of the attribute (c.f. [1]), where $(r A)$ is rounded for integer-valued attributes. How exactly it is rounded may depend on what r is and which attribute we are talking about.

Based on some—not particularly extensive—testing (but with no reference to official sources) the authors believe that the rounding happens this way (here r is the rate modifier):

- H_e modifiers are rounded toward zero (down if $r > 0$, up if $r < 0$)—tested for $r > 0$, speculation otherwise;
- Exp modifiers are applied to groups rather than individual units, and are rounded toward zero (down if $r > 0$, up if $r < 0$)—tested for $r > 0$, speculation otherwise;
- \underline{Dm} and \overline{Dm} modifiers are rounded away from zero (up if $r > 0$, down if $r < 0$)—tested for $r < 0$, speculation otherwise;
- Acc is not rounded—pure speculation;
- Spl is not rounded—pure speculation.

When there is more than one modifier, they stack as follows [1]. If a base attribute A is being modified by two modifier pairs, (x, r) and (y, s) , then

$$A + \llbracket (r + s) A \rrbracket + (x + y)$$

will be the effective value of the attribute.

The governing feature of the entire combat system is:

Assumption 1. *Units within an army attack sequentially, one at a time.*

Assumption 2. *Opposing armies attack each other simultaneously: if a unit has survived to a stage when they are supposed to attack, they are guaranteed the opportunity to attack regardless of whether or not they are killed in this stage.*

Example.

Two defending units—first with 4 hit points, second with 6 hit points—are attacked by a group with $\overline{Dm} = 5$, $\underline{Dm} = 3$ and $Spl = 0$. We will illustrate the consequences of changing the order of attack. To that end, consider two scenarios: the attacker deals (i) high, low, low; and (ii) low, low, high.

In the first case the defender's hit points will be $(4, 6) \rightarrow (0, 6) \rightarrow (0, 3) \rightarrow (0, 0)$.

In the second case the defender's hit points will be $(4, 6) \rightarrow (1, 6) \rightarrow (0, 6) \rightarrow (0, 1)$.

Example.

Three units, u_1 , u_2 , and u_3 , attack an enemy unit v . Let the attacker have 10% damage bonus; let v have 25 hit points and damage reduction of 30%.

Suppose first two attackers deal base high damage of 13 each, and the last attacker deals base low damage of 6. The inflicted damage from u_1 and u_2 will then be

$$13 + \lceil (0.1 - 0.3) \cdot 13 \rceil = 13 - \lceil 2.6 \rceil = 13 - 3 = 10$$

each. Similarly, the inflicted damage from u_3 will be

$$6 - \lceil (0.2) \cdot 6 \rceil = 4.$$

Thus the total damage will be $(10 + 10 + 4) = 24$, and v will survive with 1 hit point left.

Note that if one dropped the assumption that units attacked sequentially, the inflicted damage would have been

$$(13 + 13 + 6) - \lceil (0.2) \cdot (13 + 13 + 6) \rceil = 32 - 7 = 25$$

instead, and v would have died.

2 To Splash or Not To Splash

Splash damage (as opposed to truncated—usual—damage) is the case when overshoot damage is spilled over to the next defending unit.

Example.

Suppose there are no damage modifiers, and u attacks v —which has only 20 hit points left—with 100 damage. In either case v will be killed. If this was a *splash damage* attack, the remaining 80 damage will be inflicted onto the next unit. If this was a *truncated damage* attack, the remaining 80 damage will be lost (discarded).

While this works nicely when damage modifiers are equal between groups, the moment damage modifiers are no longer homogeneous a big question arises: how does this difference affect overshoot damage?

The answer to this question is, perhaps, the weakest point of this work: not only is it not documented officially (according to the authors' knowledge), but it has not been tested either (partly because watchtower damage reduction has been removed from the game, and coming up with a conclusive setup seemed too challenging). In short, the following is merely a description of how the authors chose to implement it, and may have little to no relevance to the game.

Suppose a unit u attacks group G followed by group H . Let $(\underline{d}_G, \overline{d}_G)$ and $(\underline{d}_H, \overline{d}_H)$ denote the effective (after all bonuses have been applied) damage of u vs. G and H , respectively. Denote the damage ratios with

$$\underline{r} = \frac{\underline{d}_H}{\underline{d}_G} \quad \text{and} \quad \overline{r} = \frac{\overline{d}_H}{\overline{d}_G}.$$

Let ϕ be the overshoot damage when G is killed. If G was killed with a low damage splash hit, then the damage spilled onto H will be $\lfloor \underline{r} \cdot \phi \rfloor$. If the damage was high, the corresponding value will be $\lfloor \overline{r} \cdot \phi \rfloor$ instead.

3 Special Abilities and Battle Structure

In this section we briefly discuss battle or unit characteristics, modifiers, and abilities, that affect the normal course of the battle.

All modifiers to He , Exp , \underline{Dm} , and \overline{Dm} are stored separately from the base values. Effective values for He are calculated before the battle begins. Effective values for \underline{Dm} and \overline{Dm} are calculated prior to each combat round. Modifiers to Acc and Spl , on the other hand, are applied before the battle directly to the base value which is then clipped to the interval $[0, 1]$.

Before the battle starts, there are two stages:

- (i) Pre-battle bonus stage: each army's bonuses are applied.
- (ii) Pre-battle penalty stage: each army's penalties are applied.

This ordering ensures, for example, that if one army has explosive ammunition (granting friendly ranged units *attack weakest target* ability) and the other has intercept (stripping enemy units of *attack weakest target* ability), intercept will take precedence, and all of the units in the first army will *not* have *attack weakest target*.

4 Combat Sequence

In this section let He^* , \underline{Dm}^* , \overline{Dm}^* , Acc^* , and Spl^* denote the modified attributes (base values after the modifiers have been applied).

While the base damage dealt by any unit is independent of any other unit, the effective damage is not. However, if the defending unit has h hit points, it will take at least $k = \lceil h / \overline{Dm}^* \rceil$ units to eliminate it, implying that the first k attacks *will* be independent. Having said that, the course of attack is outlined in the following algorithm.

Algorithm 1. Consider a group $G = (u_1, u_2, \dots, u_n)$ attacking an army $\mathcal{A} = (v_1, v_2, \dots, v_m)$. Set $i = 1$ and $j = 1$, the (one-based) index of the attacking and defending units, respectively. Let $h = He^*(v_1)$ be the health of the first defending unit.

1. Let \overline{Dm}_j^* and \underline{Dm}_j^* be the effective high and low damage against the current defending unit, and let $k = \lceil h / \overline{Dm}_j^* \rceil$ be the least number of attacking units required to kill the defender.
2. If k is greater than $(n - i + 1)$ —the remaining units in the attacking group—update $k \leftarrow (n - i + 1)$.
3. Generate $X \sim \text{Bin}(k, Acc^*)$, the number of units dealing high damage, and let $d = \overline{Dm}_j^* \cdot X + \underline{Dm}_j^* \cdot (k - X)$.
4. Update $i \leftarrow i + k$ (point to the next attacking unit).
5. Update $(h, d) \leftarrow (h - d, d - h)$ (damage the current defending unit, and store potential overshoot damage).
6. If $h < 0$ (the current defending unit has been killed and there is still leftover damage), generate $Y \sim \text{Bin}(1, Spl_u)$, the chance of splash damage (note that it is not necessary if $h = 0$). Otherwise set $Y = 0$.
7. If $Y = 0$ (no splash), update $d \leftarrow 0$ (no overshoot damage).
8. While $h \leq 0$:
 - (a) update $j \leftarrow j + 1$ (proceed to the next defending unit);
 - (b) if $j = m + 1$, terminate: the entire army \mathcal{A} has been eliminated;
 - (c) let r denote the splash ratio factor between v_j and v_{j-1} as explained in section 2;
 - (d) update $h \leftarrow He^*(v_j)$ and $d \leftarrow \lfloor r \cdot d \rfloor$;
 - (e) update $(h, d) \leftarrow (h - d, d - h)$ (damage the next defending unit, and store potential overshoot damage).
9. If $i = n + 1$, terminate: the entire group G has attacked, and exactly $(j - 1)$ units from \mathcal{A} have been killed; the j -th unit in \mathcal{A} has only h hit points left.
10. Otherwise repeat steps 1–9.

References

- [1] Official forums. [Dev Diary] Halloween 2018, <https://forum.thesettlersonline.net/threads/25261#post287808>.