osu!mania Scoring Reference - How it Works

Jinjin

April 20, 2019

Judgement	HitValue (H_{val})	HitBonusValue (B_{val})	HitBonus (b)	HitPunishment (p)
MAX	320	32	2	0
300	300	32	1	0
200	200	16	0	8
100	100	8	0	24
50	50	4	0	44
Miss	0	0	0	∞

Table 1: Note Judgement

Mod	ModMultiplier (M_m)	ModDivider (M_d)
NoMod	1	1
Easy	0.5	1
NoFail	0.5	1
HalfTime	0.5	1
HardRock	1	1.08
DoubleTime	1	1.1
NightCore	1	1.1
FadeIn	1	1.06
Hidden	1	1.06
Flashlight	1	1.06

Table 2: Mod Multipliers and Dividers

Scoring (V1)

In osu!mania, score is divided into two parts, the base score (denoted as s_{base_n}) and bonus score (denoted as s_{bonus_n}). Here, n denotes the nth note of the map, with the first note of a map having a value $n = 1$. The total score of the nth note (s_n) is calculated as

$$s_n = s_{base_n} + s_{bonus_n} \tag{1}$$

Before introducing the formula for s_{base_n} and s_{bonus_n}, we will define a few other concepts that will hopefully help with our understanding of the scoring system. Suppose a map has N_{tot} total notes and mod multiplier of M_m and mod divider of M_d. We denote hit value, hit bonus value, hit bonus, and hit punishment of the nth note as $H_{val_n}, B_{val_n}, b_n, p_n$, respectively.

Definition 1. ***Normalization constant*** *(ϕ) is a value used to adjust the base and bonus score with respect to the maximum obtainable score (one million). It is defined as*

$$\phi = \frac{1}{2}\left(\frac{1000000M_m}{N_{tot}}\right) \tag{2}$$

Definition 2. ***Cumulative bonus factor*** *(β_n) is a value that affects the amount of bonus score for the nth note. It is defined as*

$$\beta_n = \begin{cases} 100 & n = 0 \\ \max\left\{\min\left\{\beta_{n-1} + b_n - \dfrac{p_n}{M_d}, 100\right\}, 0\right\} & \text{otherwise} \end{cases}$$

Note that β_n is limited to the values from 0 to 100.

We can now see what missing a note does to the cumulative bonus factor β_n. Recall from Table 1 that a miss judgement results in $b_n = 0$ and $p_n = \infty$. Using the definition of β_n above, we calculate:

$$\begin{aligned} \beta_n &= \max\left\{\min\left\{\beta_{n-1} + 0 - \frac{\infty}{M_d}, 100\right\}, 0\right\} \\ &= \max\{-\infty, 0\} = 0 \end{aligned} \tag{3}$$

Thus, missing a note always resets the cumulative bonus factor to 0.

We can now express base score and bonus score for the nth note as follows:

$$s_{base_n} = \frac{\phi}{320} H_{val_n} \tag{4}$$

$$s_{bonus_n} = \frac{\phi\sqrt{\beta_n}}{320} B_{val_n} \tag{5}$$

Combining equations (1) with (4) and (5) yields

$$s_n = \frac{\phi}{320}\left(H_{val_n} + B_{val_n}\sqrt{\beta_n}\right) \tag{6}$$

In layman's terms, a score received from a note is calculated as a sum of the hit value of that note and the bonus value that dynamically changes with respect to the judgement of that note. The total score (S) that a player receives from a map with N_{tot} notes can then be written as a summation over scores of individual notes:

$$S = \sum_{n=1}^{N_{tot}} s_n = \sum_{n=1}^{N_{tot}} \left[\frac{\phi}{320}\left(H_{val_n} + B_{val_n}\sqrt{\beta_n}\right)\right] \tag{7}$$

Example

A Perfect Score with NoMod

This is a short example to show that $S = 1000000$ with every note hit with a MAX judgement (we assume the player has not used any mods, which means $M_m = M_d = 1$). When every note is MAX, $H_{val_n} = 320, B_{val_n} = 32, b_n = 2, p_n = 0$ for all values of n. It is then clear that $\phi = 500000/N_{tot}$ and $\beta_n = 100$ for all n. Therefore, the total score becomes

$$\begin{aligned} S &= \sum_{n=1}^{N_{tot}} \left[\frac{\phi}{320}\left(H_{val_n} + B_{val_n}\sqrt{\beta_n}\right)\right] \\ &= \sum_{n=1}^{N_{tot}} \left[\frac{1}{N_{tot}} \frac{500000}{320}\left(320 + 32\sqrt{100}\right)\right] \\ &= N_{tot} \frac{1}{N_{tot}} \frac{500000}{320} \cdot 640 = 1000000 \end{aligned} \tag{8}$$