

# Milthm ScoreV3 AccV2 草案

Richado Wonosas

2025.8

# 目 录

<b>1</b>	<b>引言</b>	<b>3</b>
<b>2</b>	<b>术语和定义</b>	<b>3</b>
2.1	缩写表 . . . . .	3
2.2	术语表 . . . . .	3
2.3	定义 . . . . .	3
<b>3</b>	<b>ScoreV3 算法</b>	<b>4</b>
3.1	判定分 . . . . .	4
3.2	连击修正乘区 . . . . .	5
3.2.1	基础修正乘区 . . . . .	5
3.2.2	修正乘区补差 . . . . .	5
3.3	连击附加分 . . . . .	6
3.4	AP 附加分 . . . . .	7
3.5	分数计算 . . . . .	7
<b>4</b>	<b>AccV2 算法</b>	<b>7</b>

# 1 引言

本文旨在提出音乐游戏《梦见霖音 Milthm》的第三版分数计算方法及第二版准确率计算方法.

## 2 术语和定义

### 2.1 缩写表

缩写	解释
Milthm	音乐游戏《梦见霖音 Milthm》.
ScoreV3	第三版分数计算方法.
AccV2	第二版准确率计算方法.

### 2.2 术语表

术语	解释
判定点	位于判定区间内的、用作判定玩家特定操作的基准时刻的点.
击打偏差	玩家的特定操作与判定点之间的时间差, 单位为毫秒. 负值表示提前于判定点, 正值表示落后于判定点.
全部连击 (FC)	对于所有判定点, 玩家的特定操作都能够产生连击接续的判定.
全部完美 (AP)	对于所有判定点, 玩家的特定操作都能够产生收益最高的数种判定.
理论值 (MAX)	对于所有判定点, 玩家的特定操作均产生最高收益.

### 2.3 定义

**定义 2.1** (判定等级). 判定等级是多个连续时间区间的并集, 用于描述击打偏差的时间区间.

对于判定等级  $l$ , 有

$$l = \bigcup_{i=1}^n I_i,$$

其中  $I_i$  是  $\mathbb{R}$  上的一段连续区间.

**定义 2.2** (判定组). 判定组  $\mathcal{J}$  是一个集合, 其中每个元素均为一个判定等级, 同时拥有一个缺省元素  $m = \emptyset$  用于描述不产生判定的情况.

**性质 2.2.1** (互不重叠). 对于任意  $j_a, j_b \in \mathcal{J}$  且  $j_a \neq j_b$ , 有  $j_a \cap j_b = \emptyset$ .

**性质 2.2.2** (区间覆盖).  $\bigcup_{j \in \mathcal{J}} j$  是  $\mathbb{R}$  上的一段连续区间.

ScoreV3 采用的判定组  $\mathcal{J}$  如下:

判定等级	名称	击打偏差时间区间（毫秒）
$s$	Exact 判定（大 P）	$[-35, 35]$
$p$	Perfect 判定（小 P）	$[-70, -35), (35, 70]$
$g$	Great 判定（大 G）	$[-105, -70), (70, 105]$
$n$	Good 判定（小 G）	$[-140, -105), (105, 140]$
$b$	Bad 判定	$[-155, -140), (140, 155]$
$m$	Miss 判定	$\emptyset$

**定义 2.3** (判定映射). 判定映射  $J: \mathbb{R} \rightarrow \mathcal{J}$  从实数域映射到判定组:

$$J(\delta) = \begin{cases} j, & \exists j \in \mathcal{J} (\delta \in j), \\ m, & \text{其他.} \end{cases}$$

**定义 2.4** (判定点序列). 判定点序列  $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_N\}$  是一串单调不减的有穷序列, 其中每个判定点  $\sigma_i \in \mathbb{R}^*$ .

**定义 2.5** (物量). 严格来讲, 物量 (物件数量) 与判定点数量并非同一概念. 本文定义的物量 (物件数量) 为一般定义下的判定点数量.

物量  $N$  是判定点序列  $\Sigma$  的长度, 即  $N = |\Sigma|$ .

**定义 2.6** (击打偏差序列). 击打偏差序列  $\Delta = \{\delta_1, \delta_2, \dots, \delta_N\}$  是一串基于判定点序列以及用户操作得到的序列, 有  $|\Delta| = N$ , 且其中每个击打偏差  $\delta_i \in \mathbb{R}$ .

**定义 2.7** (判定序列). 判定序列  $\Xi = \{\xi_1, \xi_2, \dots, \xi_N\}$  满足: 对于每个判定  $\xi_i$ , 有  $\xi_i = J(\delta_i)$ .

## 3 ScoreV3 算法

### 3.1 判定分

**定义 3.1** (阶梯式判定分算法). 阶梯式判定分算法  $S_{judge}: \mathcal{J} \rightarrow \mathbb{R}$  是一个从判定组映射到实数域的映射.

ScoreV3 采用的阶梯式判定分算法  $S_{judge}$  如下:

判定等级	得分
$s$	100,0000
$p$	99,0000
$g$	60,0000
$n$	30,0000
$b$	15,0000
$m$	0

定义 3.2 (截至第  $n$  个判定点的判定分). 截至第  $n$  个判定点的判定分  $AS_n$  定义如下:

$$AS_n = \frac{\sum_{i=1}^n S_{judge}(\xi_i)}{n}.$$

## 3.2 连击修正乘区

### 3.2.1 基础修正乘区

定义 3.3 (连击基础分映射). 连击基础分映射  $B: \mathcal{J} \times \mathbb{N}^+ \rightarrow \mathbb{R}^3$  将判定等级  $\xi$  与物量  $N$  映射为连击基础分上限  $b_{\xi,N}$ 、连击基础分下限  $d_{\xi,N}$  判定增分  $a_{\xi,N}$ .

ScoreV3 采用的连击基础分映射  $B$  如下:

判定等级 $\xi$	连击基础分上限 $b_{\xi,N}$	连击基础分下限 $d_{\xi,N}$	判定增分 $a_{\xi,N}$
$s$	$\min(\max(\lfloor 0.24N \rfloor, 1), 192)$	0	2
$p$	$\min(\max(\lfloor 0.24N \rfloor, 1), 192)$	0	1
$g$	$\min(\max(\lfloor 0.16N \rfloor, 1), 128)$	0	0
$n$	$\min(\max(\lfloor 0.12N \rfloor, 1), 96)$	0	0
$b$	$\min(\max(\lfloor 0.10N \rfloor, 1), 80)$	0	0
$m$	$\min(\max(\lfloor 0.08N \rfloor, 1), 64)$	0	0

定义 3.4 (连击分序列). 连击分序列  $\Theta = \{\theta_0, \theta_1, \dots, \theta_N\}$  是基于判定序列  $\Xi$  及连击基础分映射  $B$  计算得到的递推序列.

ScoreV3 采用的连击分序列递推公式如下:

$$\begin{aligned} \theta_0 &= b_{s,N}, \\ \theta_i &= \min(\max(\theta_{i-1} + a_{\xi_i,N}, d_{\xi_i,N}), b_{\xi_i,N}). \end{aligned}$$

定义 3.5 (连击分上限). 连击分上限  $b_{max}$  是连击分能够达到的最大值.

在 ScoreV3 中,  $b_{max} = b_{s,N}$ .

定义 3.6 (基础修正乘区序列). 基础修正乘区序列  $\mathring{K} = \{\mathring{\kappa}_1, \mathring{\kappa}_2, \dots, \mathring{\kappa}_N\}$  描述特定判定点处的基础修正乘区. 对于第  $n$  个判定点, 基础修正乘区  $\mathring{\kappa}_n$  计算方式如下:

$$\mathring{\kappa}_n = \frac{\sum_{i=1}^n \theta_i}{n \cdot b_{max}}.$$

### 3.2.2 修正乘区补差

定义 3.7 (连击分最大回复速度). 连击分最大回复速度  $a_{max}$  是连击分在各判定等级中能够达到的最快回复速度.

在 ScoreV3 中,  $a_{max} = a_{s,N}$ .

**定义 3.8** (连击分预测序列). 连击分预测序列  $\Theta^* = \{\theta_0^*, \theta_1^*, \dots, \theta_N^*\}$  描述第  $i$  个判定点处应能够达到的最大连击分.  $\Theta^*$  与连击分序列  $\Theta$  存在如下关系:

$$\theta_i^* = \max(\theta_{i-1} + a_{max}, 0).$$

**定义 3.9** (连击分回复序列). 连击分回复序列  $P = \{\rho_0, \rho_1, \dots, \rho_N\}$  描述第  $i$  个判定点后为使连击分回复到最大值的最少判定点数量 (不包含第  $i$  个判定点自身).  $P$  与连击分序列  $\Theta$  存在如下关系:

$$\rho_i = \max\left(\left\lceil \frac{b_{max} - \theta_i}{a_{max}} - 1 \right\rceil, 0\right).$$

根据定义3.3到3.9可知, 假设  $1 \leq i, j \leq N$ , 若有  $\xi_i = \xi_j$  且  $i + \rho_i \leq N, j + \rho_j > N$ , 则会产生: 两相同判定对连击乘区的扰动在不同位置不一致. 故需要引入连击分补差数列, 消除判定位置对连击乘区的影响.

**定义 3.10** (连击分补差序列). 连击分补差数列  $\hat{\Theta} = \{\hat{\theta}_0, \hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_N\}$  用于消除判定的产生位置对于基础补正乘区产生的影响. 其具体计算公式如下:

$$\hat{\theta}_0 = 0,$$

$$\hat{\theta}_i = \begin{cases} \hat{\theta}_{i-1}, & i + \rho_i \leq N \text{ 或 } \theta_i = \theta_i^*, \\ \frac{(2 \cdot (b_{max} - \theta_i) - a_{max} \cdot (N + 1 - i + \rho_i)) \cdot (i + \rho_i - N)}{2}, & \text{其他.} \end{cases}$$

**定义 3.11** (截至第  $n$  个判定点的连击补正乘区). 截至第  $n$  个判定点的连击补正乘区  $\kappa_n$  定义如下:

$$\kappa_n = 0.4 + 0.6 \cdot \frac{\max\left(\left(\sum_{i=1}^n \theta_i\right) - \hat{\theta}_n, 0\right)}{n \cdot b_{max}}.$$

### 3.3 连击附加分

**定义 3.12** (连击判定映射). 连击判定映射  $C: \mathcal{J} \rightarrow \mathbb{B}$  是一个从判定组映射到真值集合  $\mathbb{B} = \{0, 1\}$  的映射, 其中 0 代表假, 1 代表真.

ScoreV3 采用的连击判定映射  $C$  如下:

$$C(\xi) = \begin{cases} 1, & \xi \in \{s, p, g, n\}, \\ 0, & \text{其他.} \end{cases}$$

**定义 3.13** (当前连击序列). 当前连击序列  $\Gamma = \{\gamma_0, \gamma_1, \dots, \gamma_N\}$  描述第  $i$  个判定点处的当前连击数量.  $\gamma_i$  有如下递推关系:

$$\gamma_0 = 0,$$

$$\gamma_i = \begin{cases} \gamma_{i-1} + 1, & C(\xi_i) = 1, \\ 0, & C(\xi_i) = 0. \end{cases}$$

**定义 3.14** (当前最高连击序列). 当前连击序列  $\Gamma^* = \{\gamma_0^*, \gamma_1^*, \dots, \gamma_N^*\}$  描述截至第  $i$  个判定点时的最高连击数量.  $\gamma_i^*$  有如下递推关系:

$$\begin{aligned}\gamma_0^* &= 0, \\ \gamma_i^* &= \max(\gamma_i, \gamma_{i-1}^*).\end{aligned}$$

**定义 3.15** (截至第  $n$  个判定点的连击附加分). 截至第  $n$  个判定点的连击附加分  $CS_n$  定义如下:

$$CS_n = 5000 \cdot \frac{\gamma_n^*}{N}$$

### 3.4 AP 附加分

**定义 3.16** (截至第  $n$  个判定点的 AP 附加分). 截至第  $n$  个判定点的 AP 附加分  $AP_n$  定义如下:

$$AP_n = \begin{cases} 5000 \cdot \frac{n}{N}, & \forall i \in \{1, 2, \dots, n\} (\xi_i \in \{s, p\}), \\ 0, & \text{其他.} \end{cases}$$

### 3.5 分数计算

**定义 3.17** (截至第  $n$  个判定点的得分). 截至第  $n$  个判定点的得分  $S_n$  定义如下:

$$S_n = \kappa_n \cdot AS_n + CS_n + AP_n.$$

**定义 3.18** (总得分). *ScoreV3* 的总得分  $TS = S_N$ .

**性质 3.18.1** (理论值). 当且仅当  $\forall \xi \in \Xi (\xi = s)$  时,  $TS$  取得最高得分 101,0000.

## 4 AccV2 算法

**定义 4.1** (阶梯式准确率算法). 阶梯式准确率算法  $A_{judge} : \mathcal{J} \rightarrow \mathbb{R}$  是一个从判定组映射到实数域的映射.

AccV2 采用的阶梯式准确率算法  $A_{judge}$  如下:

判定等级	准确率
$s$	100%
$p$	100%
$g$	60%
$n$	30%
$b$	15%
$m$	0%

定义 4.2 (截至第  $n$  个判定点的准确率). 截至第  $n$  个判定点的准确率  $ACC_n$  定义如下:

$$ACC_n = \frac{\sum_{i=1}^n A_{judge}(\xi_i)}{n}.$$

定义 4.3 (总准确率).  $AccV2$  的总准确率  $TA = ACC_N$ .

性质 4.3.1 (最高准确率). 当且仅当  $\forall \xi \in \Xi (\xi \in \{s, p\})$  时,  $TS$  取得最高准确率 100%.