Team-Based Glicko-2 Rating with Per-Player Performance Weighting

1 Overview

This document describes a rating system for a competitive multiplayer game (4v4 or 5v5) based on the Glicko-2 algorithm, extended with:

- team-based matchmaking (two teams per match),
- per-player performance weighting (stronger contribution for players who carry, weaker for those who underperform).

Each player maintains a Glicko-2 rating state and is updated after every match, with the magnitude of the rating change modulated by their relative performance inside their own team.

2 Player State

For each player i we maintain the following Glicko-2 parameters:

- rating R_i (e.g. initial value $R_0 = 1500$),
- rating deviation RD_i (e.g. initial value $\mathrm{RD}_0 = 350$),
- volatility σ_i (e.g. initial value $\sigma_0 = 0.06$).

Internally, Glicko-2 uses a transformed scale:

$$\mu_i = \frac{R_i - 1500}{173.7178}, \tag{1}$$

$$\phi_i = \frac{\mathrm{RD}_i}{173.7178}. \tag{2}$$

These internal parameters $(\mu_i, \phi_i, \sigma_i)$ are used for updates. After each update we convert back to (R_i, RD_i) via:

$$R_i = 173.7178 \cdot \mu_i + 1500, \tag{3}$$

$$\mathrm{RD}_i = 173.7178 \cdot \phi_i. \tag{4}$$

3 Match Structure

Each match consists of two teams:

- Team A with player set $A = \{a_1, \dots, a_n\}$,
- Team B with player set $B = \{b_1, \dots, b_m\}$,

where $n, m \in \{4, 5\}$ for 4v4 or 5v5.

The match outcome is:

- $S_A = 1, S_B = 0$ if Team A wins,
- $S_A = 0, S_B = 1$ if Team B wins,
- optionally $S_A = S_B = 0.5$ for a draw (if supported).

4 Per-Player Performance Score

To incorporate individual performance, the game provides for each player i a *performance score* p_i for the match, constructed from in-game statistics (kills, deaths, objective actions, damage, etc.).

For example, one might define

$$p_i = w_{\text{kill}} \cdot \text{kills}_i - w_{\text{death}} \cdot \text{deaths}_i + w_{\text{dmg}} \cdot \text{damage}_i + w_{\text{obj}} \cdot \text{objectiveScore}_i, \tag{5}$$

where $w_{\text{kill}}, w_{\text{death}}, w_{\text{dmg}}, w_{\text{obj}}$ are design constants.

The system only needs *relative* performance within each team, so any monotonic transformation of p_i is acceptable.

5 Performance Weighting Within a Team

For each team T (either $T = A$ or $T = B$), with player set $T = \{i_1, \ldots, i_{|T|}\}$, we compute:

5.1 Team Performance Statistics

$$\bar{p}_T = \frac{1}{|T|} \sum_{i \in T} p_i, \tag{6}$$

$$s_T = \sqrt{\frac{1}{|T|} \sum_{i \in T} (p_i - \bar{p}_T)^2 + \varepsilon}, \tag{7}$$

where $\varepsilon > 0$ is a small constant (e.g. $\varepsilon = 10^{-6}$) to avoid division by zero.

Define the performance z-score for each player $i \in T$:

$$z_i = \frac{p_i - \bar{p}_T}{s_T}. \tag{8}$$

5.2 Raw Performance Weights

Define a raw performance weight for each player:

$$w_i^{\text{raw}} = 1 + \alpha \cdot z_i, \tag{9}$$

where $\alpha > 0$ controls how strongly performance influences rating changes (e.g. $\alpha \in [0.1, 0.3]$).

Clamp the raw weight to stay within a reasonable range:

$$w_i = \min\{w_{\max}, \max\{w_{\min}, w_i^{\text{raw}}\}\}, \tag{10}$$

where typical values might be $w_{\min} = 0.5$ and $w_{\max} = 1.5$.

Thus:

- players who perform around team average have $w_i \approx 1$,
- top performers get $w_i > 1$,
- underperformers get $w_i < 1$.

5.3 Normalization of Weights

To keep the average weight for a team equal to 1, normalize the weights:

$$\tilde{w}_i = w_i \cdot \frac{|T|}{\sum_{j \in T} w_j}. \tag{11}$$

Now,

$$\frac{1}{|T|} \sum_{i \in T} \tilde{w}_i = 1. \tag{12}$$

6 Team Aggregated Rating for Glicko-2

Although Glicko-2 is defined for individual vs individual matches, we can approximate team games by treating each player's opponent as the *aggregated rating* of the opposing team.

For a team T with players $i \in T$, define its internal mean and deviation:

$$\mu_T = \frac{1}{|T|} \sum_{i \in T} \mu_i, \tag{13}$$

$$\phi_T = \sqrt{\frac{1}{|T|^2} \sum_{i \in T} \phi_i^2}. \tag{14}$$

In a match between Team A and Team B, each player in Team A will treat a single opponent with parameters (μ_B, ϕ_B), and analogously each player in Team B sees (μ_A, ϕ_A).

7 Single-Opponent Glicko-2 Update

This section recalls the standard Glicko-2 update for a player i facing a set of opponents j. In our application, each player has exactly one opponent: the aggregated opposing team.

7.1 Expected Score and Helper Function

For a player with parameters $(\mu_i, \phi_i, \sigma_i)$ and a single opponent $(\mu_{\text{opp}}, \phi_{\text{opp}})$, define

$$g(\phi_{\text{opp}}) = \left(1 + \frac{3\phi_{\text{opp}}^2}{\pi^2}\right)^{-1/2}. \tag{15}$$

The expected score for player i is

$$E_i = \frac{1}{1 + \exp\left(-g(\phi_{\text{opp}})\left(\mu_i - \mu_{\text{opp}}\right)\right)}. \tag{16}$$

7.2 Variance Term v

For a single opponent, the variance term simplifies to

$$v = \left[g(\phi_{\text{opp}})^2 E_i (1 - E_i)\right]^{-1}. \tag{17}$$

7.3 Delta Term Δ

Given the actual score s_i for player i (1 if their team wins, 0 if their team loses, 0.5 for draw), define

$$\Delta = v \cdot g(\phi_{\text{opp}}) \cdot (s_i - E_i). \tag{18}$$

7.4 Volatility Update

The volatility parameter σ_i is updated by solving a one-dimensional optimization problem. Let

$$a = \ln(\sigma_i^2), \tag{19}$$

and define an iterative scheme (e.g. Newton–Raphson) to find the new a' such that

$$f(a') = 0, \tag{20}$$

where

$$f(x) = \frac{e^x(\Delta^2 - \phi_i^2 - v - e^x)}{2(\phi_i^2 + v + e^x)^2} - \frac{x-a}{\tau^2}. \tag{21}$$

Here τ is a system parameter controlling how quickly volatility can change.

Once a' is found, the updated volatility is

$$\sigma_i' = \exp\left(\frac{a'}{2}\right). \tag{22}$$

7.5 Intermediate and Final Rating Deviation

First compute the intermediate deviation:

$$\phi_i^* = \sqrt{\phi_i^2 + \sigma_i'^2}. \tag{23}$$

Then compute the new deviation:

$$\phi_i' = \left(\frac{1}{\phi_i^{*2}} + \frac{1}{v}\right)^{-1/2}. \tag{24}$$

7.6 Updated Rating Mean

Finally, the updated mean (before performance weighting) is

$$\mu_i^* = \mu_i + \phi_i'^2 \cdot g(\phi_{\text{opp}})\,(s_i - E_i). \tag{25}$$

At this point, the standard Glicko-2 update would set $(\mu_i', \phi_i', \sigma_i') = (\mu_i^*, \phi_i', \sigma_i')$.

8 Performance-Weighted Glicko-2 Update

To incorporate individual performance, we apply the Glicko-2 update as above to obtain intermediate values $(\mu_i^*, \phi_i', \sigma_i')$ and then modify only the rating mean based on the performance weight $\tilde{w}_i$.

8.1 Definition of the Weighted Update

Let

$$\Delta\mu_i = \mu_i^* - \mu_i \tag{26}$$

be the Glicko-2 change in the internal rating mean. The final mean after performance weighting is

$$\mu_i' = \mu_i + \tilde{w}_i \cdot \Delta\mu_i. \tag{27}$$

The deviation and volatility are not scaled by performance:

$$\phi_i' \text{ as computed in the standard Glicko-2 update,} \tag{28}$$

$$\sigma_i' \text{ as computed in the standard Glicko-2 update.} \tag{29}$$

Thus, performance influences the *magnitude* of the rating change for the match but not the uncertainty parameters.

8.2 Optional Clamping

To avoid extreme rating jumps, one may clamp the resulting change:

$$\Delta\mu_i' = \mu_i' - \mu_i, \tag{30}$$

and then limit it by

$$\Delta\mu_i^{\text{final}} = \min\{\Delta_{\max}, \max\{-\Delta_{\max}, \Delta\mu_i'\}\}, \tag{31}$$

with some maximum change $\Delta_{\max} > 0$. The final rating mean becomes

$$\mu_i^{\text{final}} = \mu_i + \Delta\mu_i^{\text{final}}. \tag{32}$$

9 Full Per-Match Update Algorithm

For each match between Team A and Team B:

1. For every player i, retrieve $(R_i, \text{RD}_i, \sigma_i)$ and convert to (μ_i, ϕ_i).
2. Compute team internal parameters (μ_A, ϕ_A) and (μ_B, ϕ_B).
3. Compute performance scores p_i for all $i \in A \cup B$.
4. For each team $T \in \{A, B\}$:
   (a) Compute $\bar{p}_T$, s_T, z_i for $i \in T$.
   (b) Compute raw weights w_i^{raw}, clamp to w_i.
   (c) Normalize to $\tilde{w}_i$ so that the average is 1.
5. For each player $a_i \in A$:
   (a) Let $(\mu_{\text{opp}}, \phi_{\text{opp}}) = (\mu_B, \phi_B)$.
   (b) Let $s_i = S_A$ (team outcome).
   (c) Perform the standard single-opponent Glicko-2 update using $(\mu_i, \phi_i, \sigma_i)$, $(\mu_{\text{opp}}, \phi_{\text{opp}})$, and s_i, obtaining $(\mu_i^*, \phi_i', \sigma_i')$.
   (d) Compute $\Delta\mu_i = \mu_i^* - \mu_i$.
   (e) Set $\mu_i' = \mu_i + \tilde{w}_i \cdot \Delta\mu_i$.
   (f) Optionally clamp $\mu_i' - \mu_i$ to a maximum magnitude.
6. For each player $b_j \in B$, repeat the same procedure with $(\mu_{\text{opp}}, \phi_{\text{opp}}) = (\mu_A, \phi_A)$ and $s_j = S_B$.
7. Convert each player's updated internal parameters back to (R_i, RD_i):

$$R_i = 173.7178 \cdot \mu_i' + 1500,$$
$$\text{RD}_i = 173.7178 \cdot \phi_i'.$$

10 Design Notes

- Win/loss remains the primary driver of rating changes via the Glicko-2 core update. Performance only modulates the magnitude of this change.
- The average player on a team always has an effective weight $\tilde{w}_i$ close to 1, so the team's average rating change is close to what standard Glicko-2 would produce for a team-based approximation.
- Strongly outperforming teammates in a win leads to a larger rating gain; strongly outperforming teammates in a loss leads to a smaller rating loss. Conversely, underperformers gain less on a win and lose more on a loss.
- The performance score p_i is game-dependent and should be constructed so that roles (e.g. support vs. damage) are fairly evaluated relative to other players of the same role.