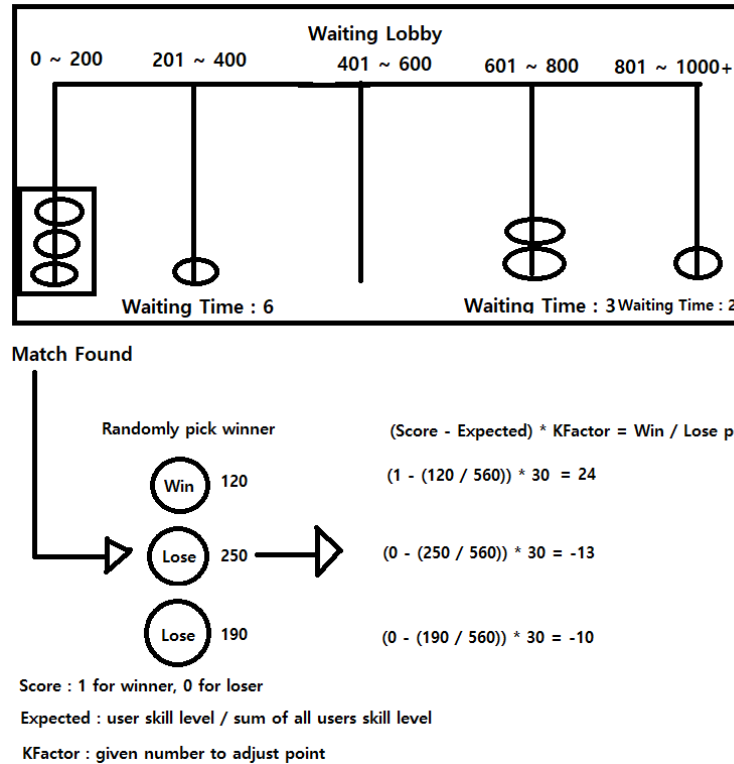


# Matchmaking Algorithms

Doosung Jang, 101175013



My algorithm uses user's skill level to match them and is divided into two parts.

The first part is Waiting Lobby. Waiting Lobby has 5 different waiting lines and each line has designated skill level range. For example, skill level 150 user will be put into the first waiting line which has skill level range from 0 to 200. In this way, all users can be matched with the users who have the similar skill level.

One problem with Waiting Lobby is that if there are not enough users who have the similar skill level, user have to wait all day in a line. To solve this problem, waiting line has waiting time. If there are any users in waiting line, waiting time counts up and then if waiting time goes beyond waiting max time, simply move users to one below waiting line to match game. In this way, users are no longer waiting all day and will find a match.

If this happens, users moved by waiting time over will be matched with users who have different skill level. This unfairness can be handled in the second part of this algorithm.

The second part is calculating win point and lose point. I slightly modified ELO rating system.

$$(\text{Score} - \text{Expected}) * \text{KFactor} = \text{win} / \text{lose point}$$

Score will be 1 for winner and 0 for loser.

Expected is calculated by  $(\text{user skill level} / \text{sum of all user skill level})$

Expected is the one who handles fairness. It is ranged from 0 to 1. In this way, if lower skill level user wins, expected will be low number and will have higher win point. Even if lower skill level user loses, expected will be low number and will have lower lose point. In contrast, if higher skill level user wins, expected will be high number and will have lower win point, and if higher skill level user loses, expected will be high number and will have higher lose point.

KFactor is just a given number to adjust number change.