osu! Rankdle HP drain

Cytusine trolling at 11pm

Suppose the player's guess is A and the actual rank of the player is B. clxxiii would like the HP lost from this guess to have the following properties:

1. The maximum HP that can be lost is 1000. (Note: based on the numbers created here, I recommend that 1000 is also the total HP given per day.)
2. HP lost should generally depend on the ratio $\frac{A}{B}$. This makes sense because pp is linearly related to log(rank), and thus this penalizes guesses based on the difference in "perceived amount of pp."
3. However, it should be more punishing to guess 10000 $\to$ 20000 than to guess 100 $\to$ 200 or 1000000 $\to$ 2000000, because proportionally more players (who are e.g. active on /r/osugame) are play the game or submit replays at that rank. Specifically, "50 $\to$ 100 should deal at most 100 HP, and 50k $\to$ 100k should deal around 1k." (This last part was not actually taken strictly into consideration.)
4. About half the maximum HP should be lost if the player guesses rank 50000 for a rank 80000 player. (This is just to set a general scale.)

Incorporating all of this information, the HP lost for a guess is

$$1000 \cdot \left[1 - \frac{\min(A,B)}{\max(A,B)}\right] \cdot \left[\frac{1}{1 + \max(0, \min[5 - \log_{10}(A), 5 - \log_{10}(B)])}\right] \cdot \left[\frac{1}{1 + \max(0, \min[\log_{10}(A) - 5, \log_{10}(B) - 5])}\right].$$

Here is the explanation for where all of this comes from – **if you don't care, just scroll to the example values on the next page**:

- The 1000 is just the maximum HP that can be lost, and each of the other three terms multiplies that by a number between 0 and 1.
- $\frac{\min(A,B)}{\max(A,B)}$ is basically the ratio between the guess and the actual rank. So if the guess is exactly correct, then that ratio is 1 and no HP is lost (because the blue bracketed term is zero). Meanwhile, guessing double or half the correct rank will deduct half the HP, guessing 10 times or 0.1 times the correct rank will deduct 90 percent of the HP, and so on.

- On the other hand, if you're guessing in the "high rank" or "low rank" regimes, meaning that **both** your guess and the actual rank are dealing with top players or relative beginners, the amount of HP lost should be much less punishing. Here, $5-\log_{10}(A)$ is basically "how many digits better than rank 100k" a player is (e.g., this value is -2 for a rank 1000 player). Since the range of values that can be guessed is between 1 and 2000000, we deal with the two cases separately.
- In the red bracketed term, the $\max(0, \min[5-\log_{10}(A), 5-\log_{10}(B)])$ part in the denominator counts, **when both A and B are smaller than 100k**, how many "digits" away from being 100k you are. (So if you're guessing rank 500 for a rank 1000 player or vice versa, this number is 2.) You then lose less HP if this number is big. (For example, in the case above, your HP lost would be cut by a factor of $\frac{1}{3}$.)
- Similarly in the green bracketed term, you are penalized less if both A and B are bigger than 100k. For example, if you're guessing rank 1 million for a rank 1.3 million player or vice versa, your HP lost will be cut by half.

Anyway, here are some example values:

Guess (A)	Rank (B)	HP Lost
100	300	189
400	300	74
1000	300	233
10000	300	485
1000	7000	398
6000	7000	66
25000	7000	449
100000	7000	930

Guess (A)	**Rank** (B)	**HP Lost**
20000	50000	461
80000	50000	342
100000	50000	500
500000	50000	900
15000	400000	963
200000	400000	384
500000	400000	125
1000000	400000	375

Here is the same formula in Python:

```
>>> import math
>>> def hp_lost(x, y):
        blue = 1 - min(x, y) / max(x, y)
        red = 1 / (1 + max(0, min(5 - math.log(x, 10), 5 - math.log(y, 10))))
        green = 1 / (1 + max(0, min(math.log(x, 10) - 5, math.log(y, 10) - 5)))
        return 1000 * blue * red * green
```