

CCC의 Elo Rating: 산출방법

목차:

1 Elo Rating을 산출의 얘기를 하기 전에:

2 새로운 순위제도 고안:

3 Elo Rating계산법:

4 순위에대한 Elo Rating 차등적용:

Elo에대한 소스코드는 프로젝트의 Capstone-Design/server/util/elo.js 파일에 존재합니다.

1.Elo Rating을 산출의 얘기를 하기 전에:

5조 팀원은 예를 들어 8명이 플레이하는 게임에서 1명만 정답을 맞추고, 바로 다음 라운드가 진행되는 것은 나머지 7명의 참가자에 대한 게임의 재미를 불러일으킬 수 없을 뿐 더러, 허무감을 불러일으킬 수 있다고 판단하였습니다. 따라서 최대한 많은 사람이 정답을제출할 수 있는 기회를 제공하면서, 게임이 지루해지지 않도록 하는 방안을 생각해냈습니다. 다음은 그 방안에 대한 소개입니다.

2.새로운 순위제도 고안:

2명	3명	4명	5명	6명	7명	8명
1명	1명	2명	2명	2명	3명	3명

위의 표는 (방의 인원 : 라운드당 정답을 제출할 수 있는 최대 인원수)의 표입니다. 방인원이 2명인 1:1 상황에서는 1명이 먼저 정답을 맞추면 라운드가 끝나는 것이 자명합니다. 방인원이 3명이 상황에서 1명으로 한 이유는 2명이 정답을 제출하면 과반수가 넘기 때문이다. 이는 경쟁의 취지와 맞지 않습니다. 4명의 방인원을 2명의 정답자를 허용하는 최소의 방 인원으로 하기로 하였습니다. 5명의 경우 그대로 2명의 정답자를 허용하기로 하였습니다. 6명의 방인원을 2명의 정답자를 허용하는 최대의 방 인원으로 하기로 하였습니다. 6명보다 많은 경우, $\text{ceil}(\text{방인원}/\text{정답인원}) = 2$ 로 만드는 수로 정답인원을 조정하였습니다.

3.Elo

Rating계산법:

elo Rating의 산출방법은 문제를 푼 사람과 문제를 풀지 못한 사람과의 "평균"으로 계산합니다.

예: 5명의 방인원에서 2명이 맞추었다면, 승자는 패자의 평균으로 Elo Rating을 계산해서 원래 elo에 더합니다.

승리자의 Elo Rating = Elo_Rating_calculate(승리자의 Elo Rating, 패자의 Elo Rating 평균) 입니다. 이것을 승리자의 사람수만큼 반복합니다.

패자의 경우도 마찬가지로 합니다.
패자의 Elo Rating = Elo_Rating_calculate(패자의 Elo Rating, 승리자의 Elo Rating 평균)
이런 계산이 나오게 된 연유에 대한 대답은, "누가 누구를 이겼나" 라는 물음에서 시작됩니다.

3.1 실제로 일어나는 문제의 예시

정답자중에 1000과, 2000의 Elo Rating을 가진 사람이 있고 정답을 맞히지 못한 사람 중에 2000의 Elo Rating을 가진 사람이 있다면, 2000의 Elo Rating을 가진 정답을 맞히지 못한 사람은 누굴 기준으로 Elo rating을 매겨야 하는가에 대한 질문이 생깁니다. 두 명 다에게 진 것을 맞지만, 그렇다고, 두 번 Elo Rating을 계산할 수는 없는 노릇입니다. 그렇게 되면 게임에서 두 번 지게 된 꼴이 되기 때문입니다.

저희 CCC팀은 이를 평균으로 매겨서 계산하자고 하였습니다. 자신이 2000의 Elo Rating을 가지고 있고, 정답자중에 1000과, 2000의 Elo Rating을 가진 사람이 있다면, 당신은 1500의 Elo Rating을 가진 사람에게 진 것으로 간주되는 것으로 하게끔 하였습니다. 저희 CCC팀은, 이 방법이 계산하기 쉽고, 직관적이라고 생각하였습니다.

4.순위에대한 Elo Rating 차등적용:

만약 당신이 들어간 방이 최대인원 8명의 방이여서, 3명의 정답자를 허용하는 방이었고 거기서 당신이 3등을 하였다면, 당신은 "승자" 이지만 원래 얻을 Elo Rating의 0.216배를 얻게 됩니다.

이유는,

정답자 1등은 원래 얻어야 할 Elo Rating의 1배를 받고,
2등은 원래 얻어야 할 Elo Rating의 0.6배를 받고
3등은 원래 얻어야 할 Elo Rating의 0.216배를 받기 때문입니다.

**2등, 3등은 원칙적으로 “승자”가 맞지만 1등과 같은 취급을 할 수 없습니다.
따라서 Elo Rating의 차등적용을 고안하게 되었습니다**

0.6이라는 숫자가 나온 이유는, 0.9, 0.8, 0.7배는 그 증감이 미미해서 2등, 3등에 대한 차이가 눈에 띄지 않기 때문에 0.6으로 정하게 되었습니다.

```
//이긴사람들 계산
for(let i=0,j=1; i<winplayer.length; i++,j = j*0.6){
    let NewRating = Elorating.calculate(getEloById(Object.values(winplayer[i])[0]),defeatAvg,true).playerRating;
    let difference = parseFloat(NewRating) - parseFloat(getEloById(Object.values(winplayer[i])[0]))
    let adjustedDifference = Math.floor(difference*j)
    result.push({[Object.values(winplayer[i])[0]]:adjustedDifference})
}
//진사람들 계산
for(let i=0; i<defeatPlayer.length; i++){
    let NewRating = Elorating.calculate(Object.values(defeatPlayer[i])[0],winnerAvg,false).playerRating;
    let difference = parseFloat(NewRating) - parseFloat(Object.values(defeatPlayer[i])[0])
    result.push({[Object.keys(defeatPlayer[i])[0]]:difference})
}
```

위의 코드는 Elo Rating계산 코드의 핵심부분입니다.
이긴 사람은 Rating을 구하고, 더할 Rating*(0.6^자신의 순위) 값을 adjustedDifference에 저장합니다.