Elo-Zahlen berechnen

Description

In einigen Spielen kann es wichtig sein, eine Elo-Zahl einzuführen. Dies gilt nicht nur für Schach. Vor allem bei Mehrspieler-Games ist dies nützlich, um gleichwertige Gegner zu finden. Dieses kurze Tutorial stellt dabei zwei Funktionen vor.

Einen ausführlichen Artikel über die Elo-Zahl gibt es auf Wikipedia, weshalb ich die Hintergründe hier nur kurz beschreiben möchte. Dabei handelt es sich um eines von mehreren Systemen (so gibt es bspw. noch Glicko und DWZ) mit denen man die Spielstärke verschiedener Spieler berechnen und vergleichen kann. Dadurch ist es auch möglich die Gewinnchancen auszurechnen. Am bekanntesten ist dieses System vom Schach, kann aber auch auf die meisten Spiele übertragen werden, vor allem, wenn die Ausgangslage immer gleich ist (wie Dame oder Go) oder die Differenzen keinen so großen Unterschied macht wie die eigentliche Fähigkeit des Spielers (bspw. Autorennen).

Gerade bei solchen Online-Spielen ist es nicht einmal nötig, dass der Spieler davon Kenntnis hat. Die Zahlen können verwendet werden, um gleichstarke Gegner zu finden. Entweder direkt über die Differenz (z. B. nicht weiter entfernt als +- 100 Punkte) oder die Siegchance, welche ebenfalls in einem bestimmten Range liegen sollte. Wer selbst ein wenig herumrechnen will, kann dieses Tool nutzen.

Die Basisformeln finden sich im bereits verlinkten Wikipedia-Artikel, aber wie wendet man sie in GML an?

Chancen berechnen

Diese Funktion besteht letztlich nur aus einer Zeile:

```
function give_chances_to_win(elo1, elo2)
{
 return round(1/(1+power(10,(elo2-elo1)/400)) * 100);
}
```

Die Funktion gibt zurück, wie hoch die Chancen in Prozent stehen, dass Spieler 1 (*elo1*) gegen Spieler

2 (*elo2*) gewinnt.

In der Praxis könnte es so aussehen, dass eine Paarung zu Stande kommt, wenn der Wert zwischen 30 und 70 steht. Viel weiter sollte es nicht auseinander liegen, da sonst beide Spieler nicht viel Spaß daran haben werden. Es ist nützlich, mehrere Spieler in eine Liste aufzunehmen, die diese Bedingung erfüllen, um dann den Spieler auszuwählen, der am ehesten bei 50% liegt.

Elo erhalten

Nun wissen wir, wie wir Chancen berechnen, aber wir brauchen zunächst überhaupt mal Werte. Neue Spieler beginnen mit einem Startwert. Im Laufe mehrerer Spiele wird der Spieler steigen oder fallen. Je nach Spiel und Plattform kann dieser Startwert sehr unterschiedlich ausfallen. Wenn man sich auf einer Schach-Plattform registriert, kann der Startwert, je nach Plattform, bei 800 bis 1500 Elo liegen (bzw. wird hier meist das Glicko-System verwendet). Als Startwert klingt ein Wert von 1000 ziemlich gut. Dies lässt viel Luft nach oben und unten. Nach jedem Spiel wird die Zahl neu berechnet. Nehmen wir an, wir gewinnen unser erstes Spiel gegen einen Spieler mit 1100 Elo (Gewinnchance 36%), dann haben wir bei einem k-Faktor (dazu gleich mehr) von 25 einen neuen Wert von 1016.

Der **k-Faktor** ist eine Gewichtung. Je größer er ist, umso stärker ist die Elo-Änderung. Wird die Elo-Zahl nur intern verwendet, spielt dies keine große Rolle. Wenn sie der Spieler sieht, kann ein höherer Faktor bei Siegen motivierender sein, ist allerdings bei Niederlagen frustrierender. Meistens liegt die Zahl zwischen 10 und 40, m. M. n. ist ein Wert von 25 oder 30 ziemlich gut. Jetzt aber zur eigentlichen Funktion:

```
function give_elo(elo1, elo2, res, factor)
{
 // res 1 = win; 0 = draw; -1 = loss
 var percentage = 1/(1+power(10,(elo2-elo1)/400));

 if (res == 1)
 {
  return round(elo1 + factor*(1-percentage));
 } else if (res == -1) {
  return round(elo1 + factor*(0-percentage));
 } else {
  return round(elo1 + factor*(0.5-percentage));
 }
}
```

Ja, dass war es schon. An die Funktion geben wir die beiden Elo-Werte, das Resultat und den Faktor weiter. Dann berechnen wir die Chancen, wobei wir nicht mehr, wie bei der ersten Funktion, mit 100 multiplizieren. Wir erhalten also eine Zahl unter 1.

Das Resultat geben wir entweder als 1 für Sieg, -1 für Niederlage oder 0 bei Unentschieden weiter. Anschließend berechnen wir, je nach Resultat, die neue Zahl.

Wichtig: Auch hier erhalten wir lediglich den Wert von Spieler 1. Wir müssen also die Funktion zweimal aufrufen, um für beide Spieler einen Wert zu berechnen.

Schlechtere Alternative

Man könnte auf die Idee kommen, sich den Spaß mit Elo zu sparen und stattdessen Siege, Niederlagen und Unentschieden zu verwenden. So könnte man die drei Resultate in einer Datenbank speichern, was man bei solchen Spielen für die Statistik ohnehin macht, zudem die Anzahl der gespielten Spiele. Gibt man Siegen 2 und Unentschieden einen Punkt, kann man sehr leicht das Verhältnis erreichter Punkte zu tatsächlichen Punkten ausrechnen.

Beispiel:

- Siege: 5
- Unentschieden: 3
- Niederlagen: 6

Macht 14 Spiele. Mögliche Punkte: 28, erreichte Punkte 13. Macht ein Verhältnis von 13/28, also 0,46. Jetzt könnte man diese Punktzahl vergleichen, um gleichstarke Gegner zu finden.

Doch leider gibt es hier einen Denkfehler: Die Zahl sagt nicht wirklich etwas über die Spielstärke aus. Das Problem gibt es am Anfang bei Elo auch, aber mit jedem weiteren Spiel „festigt" sich die Wertung, d. h. sie entspricht immer mehr der tatsächlichen Spielstärke. Bei so einem Verhältnis hingegen nicht. So könnte ich nach 1000 Spielen einen Wert von 0,46 haben und erhalte daraufhin einen Gegner mit 0,5. Dieser hat aber nur zwei Spiele gemacht, eines gewonnen und eines verloren. Es ist sehr unwahrscheinlich, dass dieser Gegner gleich stark ist.

Außerdem gilt es zu berücksichtigen, dass diese „Alternative" kaum eine Entwicklung zulässt. Je mehr Spiele ich spiele, umso weniger wirken sich die Resultate auf das Verhältnis aus. Bei Elo hingegen spielt dies kaum eine Rolle.

Date Created
9. September 2021
Author
sven