# Alapszámítás

Az értékszámításnak két fontos paramétere van:

* („score multiplier”), azt befolyásolja egyenlő értékszámok esetén a szerzett pontok hányszorosát kapja meg a játékos értékszámként.
* („spread”), az átlagértékszámtól -val való eltérés azt jelenti, hogy a játékos játékonként átlagosan pontot szerez.

Tehát:

ahol:

* a játékos értékszáma
* a játékos új értékszáma
* az értékszámok átlaga
* a játékos pontszáma

Abban az esetben, ha több játék összpontszámát gépeljük be, akkor a játékok számaszor frissíti az átlagpontszám alapján az értékszámot. Formálisan az új értékszám játék után , ahol:

# Részletek

## Belső értékszám

Az értékszámok kijelzett értékének könnyebb módosítása érdekében egy „belső” értékszámmal számol a program. Legyen az alap értékszám . Ekkor a belső értékszámra:

# Frissítés

Legyen a játékok „sokszorosságát”. Jelölje rendre az . játékos . játékban szerzett pontszámát, ebben a játékban a játékosok átlagértékszámát jelöljük -vel.

-t értelmezzük a függvényében. Jelölje szerinti deriváltját.

## 3 játékos

Legyen a játékosok átlágértékszáma.

Ekkor:

Belátható, hogy konstans marad (ha lesz időm, leírom), így ezen differenciális egyenletnek megoldása:

ahol -nál.

## játékos szimmetrikus

Tegyük fel minden emberhármas pontosan egy játékot játszott le. Ebben a részben legyen páronként különböző. Jelölje az játékos összpontszámát.

A 3 játékoshoz hasonlóan:

Jelölje az . játékos játékonkénti átlagpontszámát . legyen egy függvény a

## Nullösszegség

Könnyen belátható, hogy a frissítés során az értékszámok összege nem változik:

(Tudjuk ultiban .)

## Iteráció

A belső frissítési függvény hasonlóan:

Nézzük ezt meg kevés játékra:

Hasonlóan:

Tehát:

A program a hatékonyságért ezen képletet használja.