

Online Sports-Manager „Live“-Spielberechnung – Teil 1

Wie wir ja schon [angekündigt](#) haben, befinden wir uns in der frühen Entwicklungsphase eines Browsergame-Projekts, das wir (vorerst) Sports-Manager getauft haben.

An dieser Stelle soll euch nun ein kleiner Einblick gegeben werden, was für uns eine zentrale Herausforderung bei der Entwicklung jedes Online-Managers darstellt – das Live-Spiel oder grundsätzlich die Berechnung der Begegnungen in der Simulation. Dabei ist es unserer Ansicht nach egal, ob es sich um einen Fußballmanager, Basketballmanager oder andere Sportart handelt, die man zumindest in Grundzügen miteinander vergleichen kann.

Berechnungsansätze

Grundsätzlich kommen viele Möglichkeiten in Betracht die Berechnung der Begegnungen durchzuführen. Ein paar hiervon stellen wir vor und geben eine kurze Begründung, weshalb wir uns dafür oder dagegen entschieden haben.

Variante 1: Live-Berechnung mit KI

Die wohl komplexeste, rechenintensivste und zugleich fehleranfälligste Variante. Hierbei wird jeder Spieler auf dem Spielfeld einzeln betrachtet und trifft eigene Entscheidungen. Natürlich basieren diese „Entscheidungen“ auf den verschiedenen Gegebenheiten, Spielsituationen und Wahrscheinlichkeiten. Uns sind kaum Online-Manager bekannt, die diese Möglichkeit der Berechnung nutzen. Zwei Fußballmanager, deren Berechnung wohl zumindest teilweise auf dieser Methode basiert sind [Soccer-Match](#) und [GoalStar](#). Die genauen Berechnungshintergründe sind uns allerdings nicht bekannt, bis auf die Tatsache, dass C++ für die Berechnungen eingesetzt wird [\[1\]](#) und mehrere Server eingesetzt werden.

Für unser Projekt kommt diese Art der Berechnung aus mehreren Gründen nicht in Frage.

Zum Einen soll die Software später auch auf einem (vernünftig ausgestatteten) Webspaces laufen und nicht zwingend einen eigenen (oder auch mehrere) Server als Voraussetzung haben. Eine KI-Berechnung mit annehmbarer Performance wird außerdem schwer in PHP zu realisieren sein, so dass man hier auf andere Technologien wie Java, C++ o.ä. zurückgreifen müsste.

Variante 2: Einfache Vorabberechnung

Eine einfache Möglichkeit der Berechnung ist die Stärken beider Mannschaften zu vergleichen und hieraus das Ergebnis zu berechnen – eventuell gemischt mit ein paar Zufallswerten. Wie sich die Stärken der Mannschaften hierbei zusammensetzen ist nicht wichtig, es zählen ausschließlich die Gesamtwerte.

Hierbei kann man sich nun entscheiden, ob das stärkere Team immer gewinnt, oder ob mit einer Wahrscheinlichkeit gearbeitet wird.

Hier nun ein Beispiel:

Team A hat eine Stärke von 140, Team B eine Stärke von 120.

Berechnen wir den Anteil von Team A, so ergibt sich $140 / (140 + 120) = 140 / 260 = 0,538 = 53,8\%$.

Wir könnten – wenn wir nun mit Wahrscheinlichkeiten arbeiten – also sagen, dass Team A in knapp 54% aller Fälle die Begegnung gewinnt. In PHP Code ausgedrückt sähe das zum Beispiel so aus:

PHP

```
$teamA = 140;
```

```
$teamB = 120;
```

```
$anteilA = $teamA / ($teamA + $teamB);  
if(mt_rand(1, 100) <= $anteilA) {  
    // Team A gewinnt  
} else {  
    // Team B gewinnt  
}
```

Dies ist allerdings eine sehr vereinfachte Darstellung und es fehlen auch jegliche Überprüfungen auf Fehler oder mögliche Stolpersteine (was passiert, wenn $teamA + teamB == 0$ ist?). Wie das genaue Ergebnis der Begegnung in dieser Berechnung aussieht, würde man vermutlich auch an den Stärken festmachen. Je größer der Stärkenunterschied, desto größer auch der Unterschied im Ergebnis.

Für unsere Zwecke ist auch diese Art der Berechnung nicht geeignet, da sie die Werte viel zu vereinfacht betrachtet. Der bekannte WebSoccer dürfte in seiner Urform (also ohne Erweiterungen) eine erweiterte Variante dieser Berechnungen verwenden. Wenn man sich die Spiel-Ergebnis-Dateien ansieht ist schnell ersichtlich, dass ausgerechnet wird wie viele Minuten zwischen je zwei Toren und Torversuchen einer Mannschaft liegen. In einer reinen Textdarstellung für den Manager wäre es wohl zu eintönig, wenn die Tore immer in gleichen Zeitabständen fallen. Einen weiteren Nachteil, weswegen wir diese Art der Berechnung verworfen haben wird im nächsten Beispiel erwähnt.

Variante 3: Erweiterte Vorabberechnung

Um der Berechnung mehr Tiefe zu verleihen ist es ratsam nicht nur die Gesamtstärke eines Teams einzubeziehen, sondern auch die einzelnen Teambereiche – also Torwart, Abwehr, Mittelfeld und Sturm und diese dann miteinander zu vergleichen. Das heißt: *Abwehr Team A gegen Sturm Team B, Mittelfelder gegeneinander*, sowie *Sturm Team A gegen Abwehr Team B*. Zusätzlich den Sturm gegen den Torwart betrachten um die Torausbeuten zu ermitteln.

Ein zusätzlicher Erweiterungsschritt bestünde dann darin diese Bereiche auch noch in links, rechts und zentral aufzuteilen, so dass es in jedem Team 9 Bereiche plus Torwart gibt, die verglichen werden. *Linker Sturm Team A gegen rechte Abwehr Team B* usw.

Diese Art der Berechnung hat im Vergleich zu den zuvor genannten einfachen Vorabberechnungen schon den entscheidenden Vorteil, dass wichtig ist an welchen Positionen die Spieler im Team aufgestellt sind und wie das gegnerische Team aufgestellt ist. Das Ergebnis hängt also nicht mehr ausschließlich von den Werten ab, sondern auch von der Taktik des Managers.

Diese Möglichkeit bietet allerdings immer noch keine einfache Möglichkeit einen Text-Ticker einzubauen, in dem der Spielverlauf verfolgt werden kann. Daher reicht auch diese Art der Berechnung für unser Projekt noch nicht aus.

Eine noch deutlich erweiterte Form dieser Berechnung verwenden auch wir für unser Projekt. Welche zusätzlichen Erweiterungen wir in der Berechnung noch vorgenommen haben und welche Probleme zu lösen waren (und noch sind) behandeln wir im nächsten Teil dieser Beitragsreihe.

Quelle: <https://www.soeren-hentzschel.at/sonstiges/online-sports-manager-live-spielberechnung-teil-1/>

Online Sports-Manager „Live“-Spielberechnung – Teil 2

Im [ersten Teil](#) dieser Beitragsreihe haben wir euch schon ein paar Varianten dargestellt, wie eine Berechnung von Begegnungen in einem Online-Manager aussehen könnte. Hier folgt nun Teil 2.

Berechnungsansätze (Fortsetzung)

Der aktuelle Stand der Überlegungen wäre also eine Vorabberechnung, bei der jedes Team in 9 Bereiche plus Torwart eingeteilt wird und jeder dieser Bereiche mit dem Gegenpart aus dem anderen Team verglichen wird.

Variante 3a: Nochmals erweiterte Vorabberechnung

Im Grundsatz bietet die Berechnung schon vieles von dem, was wir uns vorstellen. Wir möchten jedoch dem Manager die Möglichkeit bieten noch mehr Einfluss auf das Verhalten des Teams zu geben. Daher soll er entscheiden können, wie das eigene Team ausgerichtet ist. Das heißt, ob es mehr über links, über rechts oder die Mitte spielt und wie offensiv die Spieler sich verhalten.

Auch diesen Sachverhalt kann man durch eine Erweiterung der Berechnung einbeziehen. Dies geschieht in zwei Schritten.

Zuerst die Offensiv-Einstellung.

Ein Team kann zwischen 0% und 100% offensiv spielen. Dabei legen wir fest, dass bei 0% offensiv die Defensive (also 3 der 9 Bereiche) einen Stärkenbonus von 30% erhält und die Offensive einen Abzug in gleicher Höhe. Das Mittelfeld bleibt von der Einstellung unberührt. Bei 100% offensiv drehen wir den Bonus um, so dass die Offensive 30% stärker und die Abwehr 30% schwächer wird. Wie hoch Bonus und Abzug sein sollten wird sich erst in späteren Tests der Berechnungen zeigen.

Als nächstes die Ausrichtung.

Ein Team kann auf jede Seite (links, Mitte, rechts) X – 100 Punkte vergeben. Diese werden dann in ein Verhältnis gesetzt um zu ermitteln über welche Seite wie oft gespielt wird. Hierbei sollte es eine Untergrenze für jeden Bereich geben, damit eine Mannschaft nicht ausschließlich über eine Seite spielt.

Um das Ganze besser zu verdeutlichen basteln wir uns ein kleines Beispiel.

Team A hat eine Gesamtstärke von 180, Team B ebenfalls.

Team A hat eine gleichmäßige Verteilung der Stärken auf die 9 Positionsbereiche:

- Defensive (l/m/r): 20/20/20
- Mittelfeld (l/m/r): 20/20/20
- Offensive (l/m/r): 20/20/20

Team B hat eine ungleichmäßige Verteilung, der Einfachheit halber nehmen wir nur an, dass die linke Seite etwas schwächer ist und die rechte etwas stärker, sowie das Mittelfeld etwas stärker als die anderen beiden Bereiche.

- Defensive (l/m/r): 14/19/23
- Mittelfeld (l/m/r): 19/22/27
- Offensive (l/m/r): 14/19/23

Bei einem reinen Teamstärken-Vergleich wie in der einfachen Betrachtung aus Teil 1 wären beide Teams identisch. In der ersten Erweiterung und der Unterteilung in Offensive, Mittelfeld und Defensive gäbe es schon deutliche Unterschiede im Vergleich. Interessant wird diese Unterteilung jedoch erst mit der Erweiterung um die Teamstrategien.

Schauen wir uns an was mit den Werten passiert, wenn beide Teams 100% offensiv (bei 30% Bonus / Abzug aus unserem letzten Beispiel – mit gerundeten Werten) spielen:

Team A:

- Defensive (l/m/r): 14/14/14
- Mittelfeld (l/m/r): 20/20/20
- Offensive (l/m/r): 26/26/26

Team B:

- Defensive (l/m/r): 10/13/16
- Mittelfeld (l/m/r): 19/22/27
- Offensive (l/m/r): 18/25/30

Betrachtet man hier nun die Verhältnisse der einzelnen Bereiche wird klar, wie viele Möglichkeiten schon diese (relativ kleinen) Erweiterungen der Berechnung dem Manager bieten, zudem er noch beeinflussen kann über welche der Spielfeldseiten seine Mannschaft hauptsächlich agiert.

Variante 3b: Generierung von Spielverläufen

Wie in den vorherigen Beispielen schon einmal erwähnt gibt es das Problem, dass bei einer solchen Berechnung die Anzahl der Tore und Torversuche ermittelt werden muss. Ein simpler Vergleich der Teamwerte lässt hier nicht viel Spielraum auch die übrigen Ereignisse wie zum Beispiel einen gescheiterten Vorstoß im Mittelfeld zu betrachten, so dass wir uns entschieden haben in der Berechnung noch etwas mehr ins Detail zu gehen.

Das Vorgehen in der Berechnung sei an dieser Stelle in einzelnen Schritten dargestellt, die eigentliche Programmierung im Hintergrund ist natürlich etwas umfangreicher.

Schritt 1: Ballbesitz

Um zu bestimmen, welches Team wie viel Anteil am Ballbesitz hat betrachten wir die Gesamtteamstärke nach Einbeziehen der kompletten Strategie und der so veränderten Werte für die einzelnen Bereiche. Im obigen Beispiel wären dies weiterhin 180 für Team A und 180 für Team B. Der Ballbesitz ist demnach in unserer Betrachtung 50%:50%.

Für die Grundberechnung bestimmen wir nun eine durchschnittliche Anzahl an Gesamtktionen im Spiel, die übertragen auf die Realität heißen: wie oft haben beide Teams insgesamt Ballbesitz. Um zumindest einen kleinen Zufallsbereich einzubeziehen erhöhen oder erniedrigen wir diesen Wert um ein kleines Bisschen.

Gesamte Aktionen im Spiel: 50

Beziehen wir nun das Stärkenverhältnis insgesamt mit ein:

Aktionen Team A (50%): 25

Aktionen Team B (50%): 25

Schritt 2: Ausrichtung der Aktionen

Um zu entscheiden, wie oft die Aktionen nun über die linke Seite, rechte Seite oder die Mitte ausgeführt werden nehmen wir die Teamstrategie aus dem vorherigen Beispiel zu Hilfe. Wir setzen voraus, dass die Strategien so eingestellt wurden, dass Team A zu 20% über links, zu 40% über die Mitte und zu 40% über rechts spielt. Team B ist so eingestellt, dass zu 50% über links, zu 20% durch die Mitte und zu 30% über rechts gespielt wird.

*Aktionen Team A (über links): $25 * 0,2 = 5$*

*Aktionen Team A (durch die Mitte): $25 * 0,4 = 10$*

*Aktionen Team A (über rechts): $25 * 0,4 = 10$*

Aktionen Team B (über links): $25 * 0,5 = 12,5$
Aktionen Team B (durch die Mitte): $25 * 0,2 = 5$
Aktionen Team B (über rechts): $25 * 0,3 = 7,5$

Schritt 3: Erfolg der Aktionen

Nun führen wir für jedes Team die Vergleiche mit den gegnerischen Bereichen durch und errechnen so, wie viele Aktionen den nächsten Bereich erreichen – also zum Beispiel von der Defensive ins Mittelfeld gelangen. Hierzu benötigen wir den Vergleich des passenden Bereichs, also nicht die linke Defensive gegen die linke Defensive, sondern gegen die rechte Offensive.

Beispielhaft berechnen wir nun den Wert für den Vergleich der linken Defensive von Team A mit der rechten Offensive von Team B:

$Team A (14) / (Team A (14) + Team B (30)) = 14 / 44 = 0,318 = 31,8\%$
somit erreichen 31,8% der Aktionen (5) das Mittelfeld, in unserem Fall also $5 * 0,318 = 1,59$

Vergleichen wir dies einmal mit der gleichen Aktion, diesmal allerdings über die rechte Seite, das heißt gegen die linke Offensive von Team B:

$Team A (14) / (Team A (14) + Team B (18)) = 14 / 32 = 0,438 = 43,8\%$
somit erreichen 43,8% der Aktionen (10) das Mittelfeld, in unserem Fall also $10 * 0,438 = 4,38$

Basierend auf diesem Wert würde nun der Wert berechnet, wie viele Aktionen erfolgreich bis in den Sturm gelangen. Also zum Beispiel rechtes Mittelfeld gegen linkes gegnerisches Mittelfeld. Im Beispiel ist das:

$Team A (20) / (Team A (20) + Team B (19)) = 20 / 39 = 0,513 = 51,3\%$
somit erreichen 51,3% der Aktionen aus dem Mittelfeld (4,38) den Sturm, in unserem Fall also $4,38 * 0,513 = 2,24$

In dieser Art würden dann auch die Aktionen errechnet, die zum Torversuch und schließlich zum Tor führen, wobei hier natürlich noch der Torwart eine Rolle spielt.

In der endgültigen Berechnung würde man diese Aktionen noch mit unterschiedlichen Boni versehen, da Aktionen die von der Defensive ins Mittelfeld führen häufiger erfolgreich sind als vom Mittelfeld in den Sturm. Die Kommastellen bei den berechneten Werten würden am Ende abgeschnitten um die tatsächliche Anzahl Aktionen zu bestimmen, die im Spielbericht aufgenommen werden.

Wie es mit der Berechnung weitergeht, erfahrt ihr dann im nächsten Teil der Beitrags-Reihe.

Quelle: <https://www.soeren-hentzschel.at/sonstiges/online-sports-manager-live-spielberechnung-teil-2/>

Online Sports-Manager „Live“-Spielberechnung – Teil 3

Im [zweiten Teil](#) dieser Beitragsreihe haben wir die Berechnungsansätze aus Teil 1 erweitert und an einem Beispiel einen ersten groben Spielverlauf im Online-Manager kreiert. Hier folgt nun Teil 3.

Berechnungserweiterungen (Fortsetzung)

Mit dem aktuellen Stand unserer Berechnungen können wir schon einen groben Spielverlauf abbilden. Wer hat wie viele Aktionen, über welche Spielfeldseite finden die Aktionen statt? Außerdem kann der Manager Strategien für sein Team einstellen, welche diese Werte beeinflussen und auch direkten Einfluss auf die berechneten Spielverläufe haben.

Bevor wir mit den Erweiterungen der Berechnung fortfahren, soll an dieser Stelle noch ein gravierender Unterschied zu einer „KI“-Berechnung oder auf Wahrscheinlichkeiten basierender Berechnungen hervorgehoben werden. Anstatt die Werte wie in unserem Berechnungsbeispiel aus dem vorherigen Teil der Beitragsreihe zu verwenden, könnten wir sie auch als Wahrscheinlichkeiten interpretieren.

In unserem Beispiel haben wir gesagt, dass Team A aus der Defensive über die linke Seite in das Mittelfeld vordringt. Hier noch einmal der entsprechende Ausschnitt:

$Team A (14) / (Team A (14) + Team B (18)) = 14 / 32 = 0,438 = 43,8\%$

somit erreichen 43,8% der Aktionen (10) das Mittelfeld, in unserem Fall also $10 * 0,438 = 4,38$

Stattdessen könnten wir auch sagen, dass nicht ein fester Anteil von 4,38 Aktionen das Mittelfeld erreicht sondern im Durchschnitt 4,38 von 10 Aktionen. In PHP ausgedrückt einmal die Berechnungen im Vergleich:

PHP

```
$teamA = 14;
```

```
$teamB = 18;
```

```
// Berechnung 1 (feste Anzahl):
```

```
$insMittelfeld = $teamA / ($teamA + $teamB);
```

```
// Berechnung 2 (Anteil mit Wahrscheinlichkeit):
```

```
$insMittelfeld = 100 * $teamA / ($teamA + $teamB);
```

```
if(mt_rand(1, 100) <= $insMittelfeld) // weiter mit nächstem Schritt
```

Natürlich kann man die Varianten nicht eins zu eins vergleichen, es sollte trotzdem sofort auffallen, dass die erste Berechnung bei jedem Aufruf genau das selbe Ergebnis liefert, die zweite jedoch nicht, da sie davon abhängt welcher Zufallswert ermittelt wird.

Für unser Projekt haben wir uns entschieden mit festen Werten zu arbeiten um die Ergebnisse absolut nachvollziehbar und auch (in gewissem Rahmen) wiederholbar zu machen. In gewissem Rahmen deshalb, da die Ergebnisse trotz fester Werte und Formeln von menschlichen Einstellungen wie der Teamstrategie und Spieleraufstellung abhängen.

Nun aber weiter mit dem eigentlichen Fortschritt der Berechnungen.

Ereignisse auf die Spielzeit verteilen

An dieser Stelle weichen wir die Aussage aus dem letzten Abschnitt etwas auf. Wir möchten zwar, dass die Spielergebnisse bei identischen Bedingungen auch zu identischen Ergebnissen führen, nicht jedoch dass der Spielverlauf identisch ist. Daher führen wir an dieser Stelle eine der wenigen Zufallsberechnungen ein, indem

wir die Anzahl der Ereignisse zufällig auf die Spielminuten verteilen. In abstrahiertem PHP Code sieht das dann etwa so aus:

```
PHP
$events = array();
$ereignisse = 5;
$ausgabe = array();
for($i = 1; $i <= $ereignisse; $i++) {
    $minute = mt_rand(0, 900) / 10;
    $ausgabe[$minute] = 'Ein Ereignis tritt ein';
}
ksort($ausgabe);
print_r($ausgabe);
```

Für die Zwecke eines Online-Managers ist dies nach unserer Ansicht völlig ausreichend. Dem Einen oder Anderen wird auffallen, dass auch 10tel Minuten möglich sind, dies hilft dabei die Ereignisse, die in der gleichen Spielminute auftreten zu sortieren.

Die Anzahl der verschiedenen Ereignisse haben wir in den vorherigen Schritten ja berechnet. So können wir nun angeben wie oft ein Angriff von welchem Team über welche Seite bis zu welcher Station stattgefunden hat und dies mit der obigen Methode auf die Spielminuten verteilen. Diese Ereignisse könnte man noch mit zufälligen Ereignissen ausschmücken, die keine Relevanz haben aber schön zu lesen sind. Natürlich dürfen auch Meldungen zu Anstoß, Halbzeit und ähnlichen Dingen nicht fehlen. Ereignisse wie Fouls, gelbe und rote Karten oder ähnliches werden derzeit auch noch nicht berücksichtigt – genauso wenig welche Spieler von den Ereignissen betroffen oder an ihnen beteiligt waren.

Ticker bzw. Ereignisbericht

Um die Ereignisse für den User / Manager aufbereitet darzustellen, werden die generierten Daten am besten in einer Datenbank hinterlegt. Hier sollte dann auch die Spielminute mit Dezimalstelle gespeichert werden um eine korrekte Sortierung und Darstellung zu gewährleisten.

Ein Spielbericht ist schön und gut, etwas Besonderes allerdings nicht. Interessant wird es, wenn die Meldungen in Echtzeit verfolgt werden können. Hier kann man sich mit einem Trick behelfen, der dem Manager den Eindruck gibt, dass die Meldungen erst entstehen obwohl sie bereits vollständig in der Datenbank hinterlegt wurden.

Zuerst legen wir fest, wie viele Spielminuten in einer echten Minute vergehen sollen.

Als Beispiel verwenden wir hier den Faktor 3.0, also vergehen pro Minute drei Spielminuten.

Als nächstes benötigen wir nun die genaue Startzeit des Spiels, die dann auch in der Datenbank hinterlegt werden sollte, beispielsweise 20.02.2012 10:00:00

Mit diesen Werten können wir nun über eine einfache Formel die aktuelle Spielminute berechnen: *Spielminute* = *vergangene Sekunden seit Spielstart* * *Faktor*

In der Anzeige für den User lesen wir also nur die Spielereignisse aus der Datenbank aus, bei denen wir eine Spielminute gespeichert haben, die kleiner ist als die berechnete aktuelle Spielminute.

Dies war Teil 3 unserer Beitragsreihe. In den nächsten Teilen versuchen wir dann näher auf Fouls und Karten einzugehen und ein Konzept zu entwickeln, mit dem der Manager auch *während* des Spiels die Team-Strategie ändern kann, obwohl die Ereignisse *vorausberechnet* wurden.

Quelle: <https://www.soeren-hentzschel.at/sonstiges/online-sports-manager-live-spielberechnung-teil-3/>

Online Sports-Manager „Live“-Spielberechnung – Teil 4

Im [dritten Teil](#) dieser Beitragsreihe haben wir die Berechnungsansätze erneut erweitert. Hier folgt nun Teil 4, wenn auch etwas später als ursprünglich geplant.

Berechnungserweiterungen (Fortsetzung)

Die grundsätzliche Berechnung der einfachen Spielereignisse funktioniert bereits und einen Liveticker für die Ereignisse können wir mit dem jetzigen Stand auch bereits abbilden. Doch natürlich fehlen noch einige entscheidende Dinge, die das Spiel interessant machen. Interessant wird ein Online-Manager unter Anderem auch dadurch, dass die Teams von echten Menschen gesteuert werden, sich also in der Taktik, Aufstellung und Spielweise unterscheiden.

Bei vielen „einfachen“ (im Sinne der Spielberechnung) Managern werden die Ergebnisse vorausberechnet, ähnlich wie wir es in unserem Ansatz ja auch tun. Oftmals kann man festlegen in welcher Spielminute ein Spieler gegen einen anderen ausgetauscht werden soll, das war es dann aber auch schon. Wir möchten jedoch gerne, dass ein User auch während eines laufenden Spiels noch eingreifen und seine Taktik oder Aufstellung ändern kann. Doch wie verträgt sich das mit der Vorausberechnung der kompletten Begegnung?

Strategieänderungen trotz Vorabberechnung der Begegnungen

Diese Funktion war schon in der Planung eine ziemliche Herausforderung. Wie schaffen wir es ein Spiel komplett zu berechnen, aber dennoch dem User die Möglichkeit zu geben einzugreifen – zu jeder Zeit des Spiels?

Einfacher wäre dies gewesen, wenn wir uns für einen Berechnungsansatz entschieden hätten, der das Spiel minutenweise berechnet – jede Spielminute einzeln. Da dies aus performancetechnischen Gründen nicht in Frage kam musste eine andere Lösung her. Eine besondere Herausforderung ist hierbei, dass die Ereignisse nicht gleichmäßig über die Spielzeit, sondern zufällig auf die Spielminuten verteilt werden (siehe [Teil 3](#)).

Wie im letzten Teil auch schon erwähnt wurde, können wir die aktuelle Spielminute eines laufenden Spiels jederzeit ermitteln, indem wir die Differenz der Startzeit des Spiels und der aktuellen Zeit betrachten und den Geschwindigkeitsfaktor der Berechnung berücksichtigen. Möchte ein Spieler nun seine Strategie ändern und schickt die Änderung ab, so weiß der Server ab welcher Spielminute die Änderung greifen muss.

Die Lösung

Die Lösung ist gleichermaßen simpel und kompliziert (ja, irgendwie ein Widerspruch). Das Spiel wird einfach ab dem Zeitpunkt der Strategie-Änderung neu berechnet. Alle vorher berechneten Ereignisse des Spiels werden beibehalten, alle Ereignisse ab der Änderungsminute werden gelöscht und neu berechnet. Das war der simple Teil.

Das Komplizierte ist, dass – wie oben erwähnt – die Ereignisse nicht gleichmäßig verteilt sind. Angenommen ein Team würde mit der Startaufstellung und -strategie aufgrund der Werteverhältnisse (siehe die ersten Beitrags-Teile) 3 Tore schießen. Durch die zufällige Verteilung der Ereignisse fallen diese 3 Tore jedoch erst in der 50., 69. und 80. Minute. Nun greift der User in der 60. Spielminute ein und stellt seine Aufstellung und Strategie so um, dass sein Team mit dieser Einstellung über die 90 Minuten gerechnet kein Tor schießen würde.

Würden wir die Begegnung einfach ab der 60. Minute neu berechnen, so hätte das Team am Ende nur 1 Tor geschossen. Rechnerisch wären aber bis zur 60. Minute 2 Tore geplant. Um dieses Problem zu lösen, berechnen

wir einen „Überhang“, also die Anzahl Tore (und andere Ereignisse), die zuwenig oder zuviel in den vergangenen Minuten geschehen sind.

Hier ein Beispiel, die Voraussetzungen sind wie zuvor geschildert:

1. Berechnung der Begegnung, Team schießt 3 Tore in der 50., 69. und 80. Minute.
2. User greift in der 60. Minute ein, die neuen Einstellungen würden zu 0 Toren in 90 Minuten führen.
3. Neuberechnung der Begegnung ab der 60. Minute
 - 3a. Berechnung der „fairen Tore“ bis zur 60. Minute: $3 * 60 / 90 = 2$
 - 3b. Tatsächliche Tore: 1, dadurch ein Überhang von 1 Tor.
 - 3c. Berechnung der Tore in den verbleibenden 30 Minuten: 0, Addition des Überhangs: 1 -> Tor in der 83. Minute (Zufallswert zwischen 60. und 90.)

Der gleiche Ansatz funktioniert natürlich auch in die andere Richtung. Angenommen ein Team würde in 90 Minuten mit den Starteinstellungen 1 Tor schießen und dieses fällt bereits in der 10. Minute. In der 20. Minute greift der User ein und stellt das Team so um, dass es auf 90 Minuten gerechnet 3 Tore schießen würde. Hier die Beispielrechnung anhand dieser Werte:

1. Berechnung der Begegnung, Team schießt 1 Tor in der 10. Minute.
2. User greift in der 20. Minute ein, die neuen Einstellungen würden zu 3 Toren in 90 Minuten führen.
3. Neuberechnung der Begegnung ab der 20. Minute
 - 3a. Berechnung der „fairen Tore“ bis zur 20. Minute: $1 * 20 / 90 = 0,22222$
 - 3b. Tatsächliche Tore: 1, dadurch ein Überhang von $-0,77778$ Toren.
 - 3c. Berechnung der Tore in den verbleibenden 70 Minuten: $3 * 70 / 90 = 2,33333$, Addition des Überhangs: $1,55555$ -> Tor in der 64. Minute (Zufallswert zwischen 20. und 90.)

Am zweiten Beispiel sieht man sehr schön, dass die verschiedenen Dauern, mit denen eine Einstellung berechnet wurde, berücksichtigt wird. Es fallen also im gesamten Spiel nicht 1 oder 3 Tore, sondern 2 (Komma Stellen werden abgeschnitten). Nach diesem Muster wird das Spiel nach jedem Eingreifen eines Users neu berechnet. Dabei wird das Verfahren auch für die anderen Werte angewendet, also Fouls, Torschüsse etc. Nach unserer Meinung ist dieser Weg optimal um gleichermaßen performant und flexibel zu berechnen und die Möglichkeit einer echten Interaktion zu schaffen.

Dies war Teil 4 unserer Beitragsreihe. Der nächste Teil wird sich mit der Berechnung von **Fouls**, **Verletzungen** und **Strafen** befassen.

Quelle: <https://www.soeren-hentzschel.at/sonstiges/online-sports-manager-live-spielberechnung-teil-4/>

- Ende -