

Programmierung eines Multiplayer Online Fußballmanagers mit PHP und MySQL

Fachsbereichsarbeit im Wahlpflichtfach Informatik

Patrick Stummer

Betreuer

Mag. Carl Metnitz

GRG 1 Stubenbastei
Stubenbastei 6-8, 1010 Wien

Schuljahr 2005/2006

Wien, im Februar 2006

1 Inhaltsverzeichnis

1	Inhaltsverzeichnis	- 1 -
2	Vorwort	- 2 -
3	Einleitung	- 3 -
4	PHP.....	- 4 -
4.1	Wieso PHP?	- 4 -
4.2	Entwicklung von PHP	- 6 -
4.3	Vorteile von PHP?	- 7 -
5	Mysql	- 9 -
5.1	Was ist eine Datenbank?	- 9 -
5.2	Was ist MySQL?	- 10 -
5.3	Wozu braucht ein Fußballmanager eine Datenbank?	- 10 -
6	Grundzüge des Spiels.....	- 12 -
6.1	Was macht ein Fußballmanager?	- 12 -
6.2	Zwei wichtige Schlagworte: Multiplayer und Browserbasiert.....	- 12 -
6.3	Was das Spiel alles können soll.....	- 13 -
6.4	Das Ziel bei einem Fußballmanager.....	- 14 -
7	Planung und Realisierung des Fußballmanagers	- 15 -
7.1	Die Planung – ein entscheidender Faktor	- 15 -
7.2	Das Wasserfallmodell	- 15 -
7.3	Prototyping.....	- 16 -
7.4	Das Spiralmodell.....	- 17 -
7.5	Das von mir angewandte Modell.....	- 18 -
8	Effektives arbeiten mit der Template Engine	- 19 -
8.1	Was ist eine Template Engine	- 19 -
8.2	Wieso eine Template Engine verwenden?	- 20 -
9	Objektorientierte Programmierung (OOP)	- 22 -
9.1	OOP ganz allgemein	- 22 -
9.2	Wieso OOP verwenden?	- 22 -
9.3	Grundlegende Prinzipien von OOP	- 23 -
10	Cronjobs.....	- 24 -
10.1	Das Spiel läuft von selbst.....	- 24 -
10.2	Was ist ein Cronjob?	- 24 -
10.3	Die Crontab für den Fussballmanager	- 24 -
11	Das Herzstück des Spiels – Die Spielsimulation.....	- 26 -
11.1	Die Aufgaben des AWP	- 26 -
11.2	Das Grundprinzip.....	- 27 -
11.3	Die eigentliche Simulation des Spiels	- 27 -
11.4	Die Probe	- 28 -
12	Anhang	- 31 -
12.1	Anhang A - Quellenverzeichnis.....	- 31 -
12.2	Anhang B – Links zu webbasierten Projekten	- 32 -
12.3	Anhang C – Quellcodeverzeichnis.....	- 32 -
12.4	Anhang D – Zur CD.....	- 32 -
12.5	Anhang E – Danksagung	- 33 -

2 Vorwort

Seit ich mich mit dem Computer beschäftige, kenne ich Fußballmanager. Denn da ich ja ein fußballbegeisterter Mensch bin zählten diese natürlich zu meinen ersten Spielen die ich hatte und spielte. Mit World Wide Web, lernte ich auch Multiplayer Online Fußballmanager kennen, und auch hierfür entfachte meine Begeisterung.

Das WWW übte eine derartige Faszination auf mich aus, dass ich mich immer mehr damit befasste und bald darauf hatte ich auch schon meine erste eigene Website. Es war eine ganz einfache HTML-Website, bei www.beepworld.de erstellt, wo man nur Text schreiben musste der automatisch in HTML umgewandelt wurden. Aber es war ein Anfang. Das Thema der Homepage war übrigens mein Verein bei einem Online Fußballmanager. Der nächste Schritt war das Erlernen HTML. Ich lernte die Grundlagen und kurze Zeit später hatte ich meine erste ganz selbst gemachte Website. Doch das stellte mich nicht zufrieden. Vielmehr war ich fasziniert von dynamischen Websites und im speziellen natürlich von Fußballmanagern. Ich wollte Techniken erlernen damit auch ich Dynamik in meine Website bringen konnte und hatte den Traum einmal selbst einen Fußballmanager zu programmieren.

Und so ist es dann auch geschehen. Ich kaufte mir meine ersten Bücher und Zeitschriften über PHP und machte erste zaghafte Versuche mich als Programmierer zu betätigen und nach einiger Zeit verstand ich bereits einige grundlegende Elemente von PHP. Ich fing meine Arbeit für eine Art Fußballmanager an. Allerdings kannte ich damals noch keine Datenbanken und war entschlossen, dass die Realisierung auch mit Hilfe von einfachen Textdateien zu bewerkstelligen sein müsste. Schließlich war das erste kleine Skript fertig. Zwar funktionierte es und ich konnte bereits ein kleines Turnier mit einer Hand voll Freunden spielen, aber ich verstand, dass für einen Fußballmanager für mehrere Spieler weit mehr nötig war als ich bis dahin konnte.

Also ging ich wieder dazu über weiter zu lernen. Ich musste mir noch Grundkenntnisse von Datenbanken und professionellen Webprogrammertechniken wie Template Engines oder multiple Sprachunterstützung aneignen.

3 Einleitung

In dieser Fachbereichsarbeit geht es um eine von mir entwickelte Software mit Hilfe der Skriptsprache PHP und dem Datenbankmanagementsystem MySQL. Doch kann im Rahmen dieser Fachbereichsarbeit nicht der gesamte Quellcode der von mir entwickelten Fußballmanager-Software ausführlich erklärt werden. Vielmehr geht es darum den logischen Aufbau des Programms zu erläutern. Denn wenn man den gesamten Code, der benützt wird zusammenzählt, so kommt man auf weit mehr als 20.000 Zeilen.

Im Prinzip ist diese Fachbereichsarbeit in 4 Teile gegliedert. Der erste Teil, Kapitel 1 und Kapitel 2, beschäftigt sich mit beiden Grundbausteinen des Spiels, der Skriptsprache PHP und dem Datenbankmanagementsystem MySQL. Die nächsten beiden Kapitel widmen sich der Planung, ehe der dritte Teil auf Techniken eingeht, nämlich wie die Software professionell entwickelt werden kann. Im letzten Abschnitt komme ich dann zum praktischen Teil, in dem das Auswertungsprogramm ganz genau unter die Lupe genommen wird.

4 PHP

4.1 Wieso PHP?

Kennen Sie den Unterschied zwischen statischen und dynamischen Webseiten? Statisch sind zum Beispiel reine HTML-Seiten. Sie liegen als fertiges Dokument auf irgendeinem Webserver und können bei Bedarf abgerufen werden. Das funktioniert dann so:

- Man tippt eine Adresse in die Adresszeile des Browsers ein (z.B. `www.psw-manager.de/index.html`)
- Das Hypertext Transfer Protocol sendet eine Anfrage an den Webserver
- Der Webserver lädt das Dokument und sendet sie an den Browser zurück
- Der Browser zeigt die HTML-Seite an

Das würde allerdings bei einem Fußballmanager nicht funktionieren, denn dann würde ja jeder Mitspieler nur ein und dieselbe Seite, also die gleiche Information angezeigt bekommen. Jedoch soll jeder nur Informationen zu seinem eigenem Verein erhalten, nicht aber die eines anderen. Man muss also den Inhalt an den Benutzer anpassen. Um das zu erreichen, helfen aber statische Seiten wenig weiter, denn man muss den Inhalt für jeden Benutzer neu generieren. Die Seite muss also dynamisch werden.

Dynamisch kann man eine Seite auf zwei Arten machen. Entweder man benützt eine clientseitige oder eine serverseitige Technologie. Wenn eine clientseitige Technologie zum Einsatz kommt, dann verwendet man Programmiersprachen wie zum Beispiel JavaScript oder Java. Diese (und auch die meisten anderen clientseitigen Technologien) werden direkt in HTML eingebunden. Die Dynamik der Seite beschränkt sich dabei aber zumeist auf das Anzeigeformat der Webseite, da clientseitige Techniken keinen Zugriff auf Datenbanken oder Daten, die auf dem Webserver lagern, besitzen. Dokumente, die Programmiersprachen beinhalten die clientseitig ausgeführt werden, werden nämlich vom Webserver als normale HTML-Seite interpretiert und genauso wie HTML-Dokumente unverändert an den Browser gesandt. Man kann also durch clientseitige Technologien kaum den Inhalt, sondern nur die Anzeige dynamisch gestalten. Ein weiterer Nachteil ist, dass die Skripte erst im Browser ausgeführt werden können, und nicht jeder Browser alle

Programmiersprachen unterstützt. Außerdem besitzen fast alle Browser die Möglichkeit clientseitige Programmiersprachen abzuschalten. Fazit ist, dass auch diese Methode für einen Fußballmanager unzureichend ist.

Bleibt noch die zweite Art Webseiten dynamisch zu gestalten, nämlich die serverseitigen Technologien, zu denen auch PHP gehört. Bei solchen Technologien läuft die Anforderung an den Webserver ein kleines bisschen anders als bei den clientseitigen Technologien.

- Man tippt eine Adresse in die Adresszeile des Browsers ein (z.B `www.psw-manager.de/index.php`).
- Das Hypertext Transfer Protocol sendet eine Anfrage an den Webserver
- Anhand der Dateiendung `.php` (oder einer alternativen Dateiendung von PHP) erkennt der Webserver, dass es sich um eine PHP-Datei handelt, und lädt diese in das PHP-Modul.
- Zuerst überprüft der Parser¹, ob der Code syntaktisch korrekt ist. Wenn das nicht der Fall ist, wird eine Fehlermeldung an den Browser zurückgesandt, ansonsten wird der Code ausgeführt und in das gewünschte Format (bei einer Webseite meistens HTML) übersetzt.
- Der Webserver schickt die übersetzte Seite an den Browser.
- Der Browser zeigt nun die gerade generierte HTML-Seite an.

Wie man sieht und wie der Name auch schon verrät, laufen serverseitige im Unterschied zu clientseitigen Technologien auf dem Webserver und nicht im Internetbrowser ab. Das heißt, dass die Programme oder Skripte ausgeführt werden bevor sie an den Browser gesandt werden. Das bringt eine Reihe von Vorteilen mit sich:

- Die verwendete Programmiersprache muss nur von dem Webserver unterstützt werden auf dem die Datei liegt. Der Besucher hat keine Chance die Skripte über den Browser abzuschalten.
- Dadurch, dass die Datei direkt auf dem Webserver ausgeführt wird, hat das Programm oder Skript auch Zugriff auf alle auf dem Server lagernden Daten und etwaige Datenbanken.

¹ Der Parser überprüft den PHP-Code auf die syntaktische Korrektheit

- Durch serverseitige Technologien kann man bei jeder Anfrage den Inhalt neu generieren, bevor die Seite an den Browser gesandt wird. Es ist daher möglich sich an den Besucher anzupassen.

4.2 Entwicklung von PHP

Es war Rasmus Lerdorf, ein Däne, der 1995 PHP entwickelte. Ursprünglich stand PHP dabei für „Personal Home Page Tools“ und war eine Ansammlung an Perl-² Skripten, welche die Aufgabe hatten, die Zugriffe auf Lerdorfs Onlineauftritt zu protokollieren. Wenig später jedoch schrieb er eine größere Umsetzung in C³, jene Programmiersprache in der PHP auch heute noch geschrieben ist. Diese erste Version von PHP nannte er PHP/FI (Personal Home Page Tools / Form Interpreter). 1997 erschien dann PHP/FI 2.0. Es war eine Überarbeitung der ersten Version und schaffte es bereits sich bei geschätzten 50.000⁴ Benutzern durchzusetzen, obwohl PHP zu der Zeit noch ein Ein-Mann-Projekt war.

Schließlich kam PHP 3, die erste Version, die man mit dem heutigen PHP vergleichen kann. Andi Gutmans und Zeev Suraski beschlossen, das bestehende PHP/FI 2.0 komplett zu überarbeiten, da es ihrer Meinung nach für die Entwicklung ihrer eCommerce-Applikation viel zu schwach war. Sie entschieden sich, aufbauend auf die bestehenden PHP/FI-Benutzer, zur Kooperation mit Rasmus Lerdorf und kündigten PHP 3.0 als offiziellen Nachfolger von PHP/FI 2 an. Neben der Skriptsprache an sich wurde auch der Name überarbeitet. Seither ist PHP ein rekursives Akronym und steht für „PHP Hypertext Preprocessor“. Allerdings bereits kurz nach der Veröffentlichung von PHP 3 begannen Andi Gutmans und Zeev Suraski wieder den Kern von PHP umzuschreiben. Ihr Ziel war es diesmal komplexe Applikationen effizienter behandeln zu können. Diese neu entwickelte Engine nannten sie ‚Zend Engine‘ (aus den Vornamen Zeev und Andi gebildet). Sie wurde zum ersten Mal bei PHP 4 eingesetzt.

² eine Programmiersprache

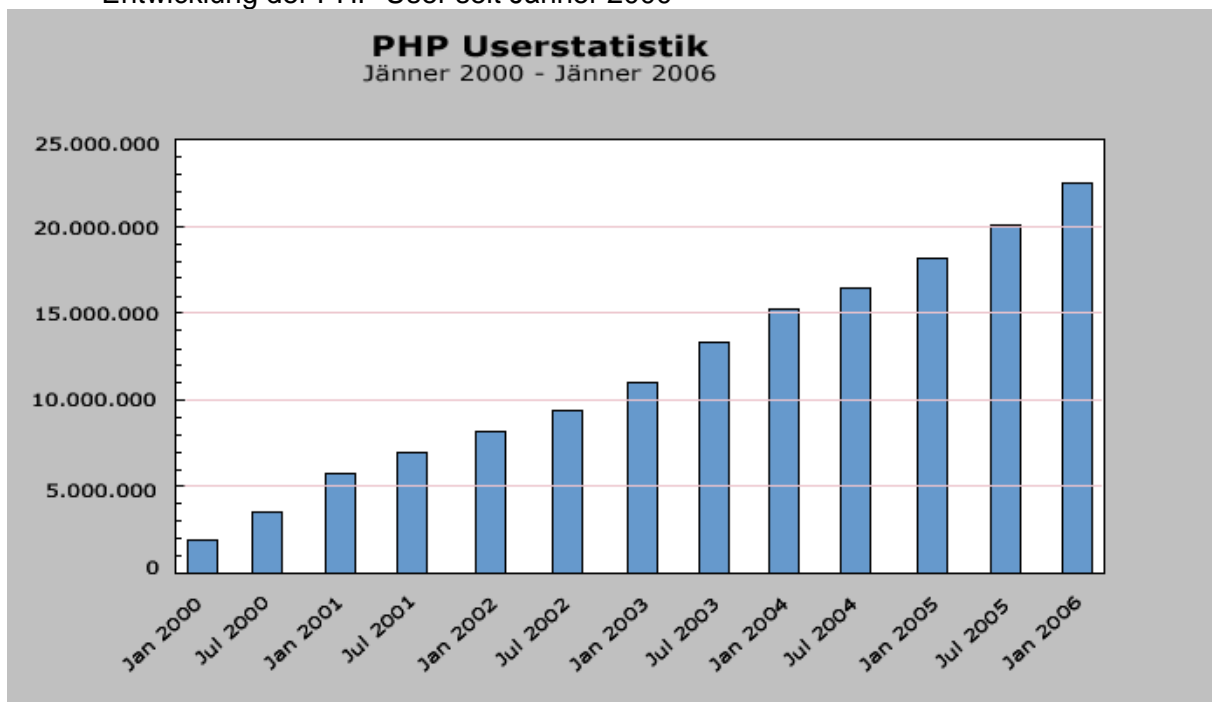
³ ebenfalls eine Programmiersprache

⁴ Statistik: www.php3.de

Am 13. Juli 2004 erschien die bisher aktuellste Version von PHP, die Version 5. Sie läuft unter der überarbeiteten Zend Engine 2 und brachte vor allem Verbesserungen was das Objektorientierte Programmierung (OOP) betrifft. PHP 6 ist für das dritte Quartal 2006 geplant.

Das PHP einmal ein Ein-Mann-Projekt war, kann man heute kaum mehr nachvollziehen. PHP ist zu einer der führenden Programmiersprachen im Web herangewachsen und wird bereits von fast 25 Millionen Usern genützt. Und der Trend zeigt immer noch stark nach oben.

Entwicklung der PHP User seit Jänner 2000⁵



4.3 Vorteile von PHP?

PHP ist eine Skriptsprache deren Syntax stark an die von C# oder Java erinnert. Doch im Unterschied zu vielen anderen Programmiersprachen hat PHP viele entscheidende Vorteile vor allem im Zusammenhang mit der Webprogrammierung.

⁵ Statistik: www.php.net; Grafik: eigene Grafik

- PHP wurde speziell für das Web entwickelt. Der große zur Verfügung stehende Funktionsumfang ist in erster Linie auf Problemstellungen ausgerichtet welche bei der Webprogrammierung immer wieder auftreten.
- PHP ist im Vergleich zu anderen Programmiersprachen verhältnismäßig leicht zu erlernen.
- PHP wird durch spezielle Tags in den HTML-Code integriert. Statische Inhalte müssen daher nicht wie zum Beispiel in Perl umständlich durch Ausgabebefehle erzeugt werden.
- PHP ist eine Open Source-Software. Es ist auf fast allen wichtigen Webservern installiert und praktisch für alle gängigen Betriebssysteme und Server gratis erhältlich.
- PHP Skripte sind nicht plattformabhängig. Man kann Skripte die man auf einem Windows-System geschrieben hat meist problemlos auf Linux-Systemen laufen lassen und umgekehrt.
- PHP ist gut skalierbar, kann also auch eine große Anzahl an Anforderungen (bei vielen Besuchern) gleichzeitig abwickeln ohne wesentlich an Geschwindigkeit zu verlieren.

PHP ist eine dynamisch getypte Skriptsprache. Das heißt Variablen von PHP sind nicht typenspezifisch gebunden wie das bei statisch getypten Programmiersprachen wie C der Fall ist. Man kann also in PHP einer Variablen, die einen String-Wert beinhaltet problemlos einen Integer-Wert zuweisen, ohne dass dies zu Problemen führt. Denn die Variable ist nicht typenspezifisch gebunden. Anders als bei C, wo es zu einem Fehler kommt, wenn man die oben genannte Aktion ausführt, da man in C mit der Initialisierung der Variable auch den Typ festlegen muss und diesen im Nachhinein nicht mehr ändern kann.

5 Mysql

5.1 Was ist eine Datenbank?

Im Prinzip könnte eine Datenbank (engl. database) aus einer einfachen Datei in der Daten gespeichert sind bestehen. Allerdings meint man, wenn man von einer Datenbank spricht neben den gespeicherten Daten meist auch das Datenbankmanagementsystem (DBMS). Dieses hat die Aufgabe, die gespeicherten Daten zu organisieren und die Kommunikation zwischen dem Benutzer und der Datenbank zu ermöglichen. Dazu bedienen sich die meisten Datenbankmanagementsysteme einer eigenen Sprache. So auch MySQL welches sich der weit verbreiteten Structured Query Language (SQL) bedient.

SQL ist aus SEQUEL (Structured English Query Language) hervorgegangen. Ursprünglich war SQL als SEQUEL/2 gedacht, jedoch musste man den Namen aus rechtlichen Gründen in SQL umbenennen. SEQUEL wurde in 1970er von IBM entwickelt, SQL kam 1976 auf den Markt. SQL hat eine sehr einfache an die Englische Umgangssprache angelehnte Syntax. Man kann sie in vier eigenständige Sprachabschnitte aufteilen, denen unterschiedliche Aufgaben zukommen.

„1. Data Definition Language (DDL) zur Definition des Datenbankschemas (Satzbeschreibung).

2. Data Query Language (DQL) zur Datenabfrage.

3. Data Manipulation Language (DML) zur Datenmanipulation (Änderungen, etc.).

4. Data Control Language (DCL) zur Kontrolle der Datenbank-Ressourcen.“

(Zitat aus Wikipedia⁶)

SQL wird von vielen bekannten Datenbanksystemen unterstützt und hat sich deswegen als eine Art Standardsprache der Datenbankkommunikation etabliert.

Natürlich könnte man auch mit einem selbst geschriebenen PHP-Skript Daten in eine Datei schreiben und dort organisieren, allerdings ist das wenig sinnvoll. Es gibt eine

⁶ <http://de.wikipedia.org/wiki/SQL>, 4.1.2006

Menge bestehender Datenbankmanagementsysteme die dahingehend optimiert sind einen schnellen und einfachen Zugriff auf die gespeicherten Daten zu ermöglichen. Diese haben meist einen großen Funktionsumfang und viele Programmiersprachen bieten eigene Techniken um den Zugriff auf bestimmte Datenbanksysteme zu ermöglichen (zum Beispiel MySQL-Funktionen in PHP). Außerdem sind viele dieser Datenbanksysteme Open Source-Software und können daher gratis herunter geladen und benutzt werden.

5.2 Was ist MySQL?

Die meisten Datenbanken organisieren ihre Daten in Tabellenform (Relationale Datenbanksysteme; Beziehung zwischen 2 Tabellen ist die Relation). Aus diesen Tabellen und den Relationen zwischen den einzelnen Tabellen ergibt sich dann die Struktur (oder Schema) der Datenbank. Bei solchen Datenbanken spricht man von einem relationalen Datenbanksystem. Neben diesem System existieren noch einige andere Datenbankkonzepte wie zum Beispiel XML, wo die Daten in einer Tag-Struktur gespeichert werden, oder Datenbanken, die auf einer Objektstruktur basieren. MySQL gehört zur Gruppe der relationalen Datenbanksysteme.

Kurz gesagt ist MySQL ein SQL-Datenverwaltungssystem. Das bedeutet soviel, dass es die Befehlssprache SQL (Structured Query Language) verwendet um mit der Datenbank zu kommunizieren. MySQL unterstützt beliebig viele Datenbanken und beliebig viele Tabellen, die Obergrenze bestimmt hier das Betriebssystem.

5.3 Wozu braucht ein Fußballmanager eine Datenbank?

Wie bereits im vorhergehenden Punkt erwähnt, werden in Datenbanken in irgendeiner Form Daten langfristig gespeichert. Das ist dann sinnvoll, wenn man irgendwelche Daten, die entweder vom Besucher oder einem Skript an den Webserver gesendet werden, später benötigt oder dauerhaft archivieren will.

Bei einem Fußballmanager ist das in vielen Situationen der Fall. Auf der einen Seite ist es wichtig Daten, die der User an den Webserver sendet, wie zum Beispiel die

Aufstellung oder Taktik, zu speichern und für das Auswertungsprogramm bereit zu halten. Auf der anderen Seite ist es notwendig auch die Ergebnisse der Spiele, wie sie vom AWP berechnet werden, zu archivieren und nicht zuletzt müssen in irgendeiner Form Informationen zu den Mitspielern, den virtuellen Fußballspielern und virtuellen Fußballklubs gespeichert werden.

6 Grundzüge des Spiels

6.1 Was macht ein Fußballmanager?

Sucht man im World Wide Web oder in den Regalen eines Computerspielladens so wird man einige Produkte finden, die sich als Fußballmanager deklarieren. Doch was ist das nun eigentlich genau?

Ein Fußballmanager ist ein Spiel, bei dem der Benutzer in die Rolle eines virtuellen Managers (und meistens auch eines virtuellen Trainers) eines Fußballvereins schlüpft und einen solchen Verein leiten (und trainieren) darf. Der Fußballmanager versucht dabei den Alltag des Managers und die Probleme der Vereine möglichst realistisch zu simulieren und dem Benutzer möglichst großen Spielspass zu ermöglichen. Zu dieser realistischen Simulation gehört, möglichst viele Manager- und Trainertätigkeiten nachzuahmen. Zum Beispiel das Leiten des Trainings, das Setzen der Aufstellung für das nächste Spiel, die Gehaltsverhandlungen mit den Spielern oder auch Transferverhandlungen mit anderen Vereinen und noch vieles mehr. Mehr dazu dann in Kapitel 6.3.

6.2 Zwei wichtige Schlagworte: Multiplayer und Browserbasiert

Mein Fußballmanager ist ein so genanntes Massive(ly) Multiplayer Online Game (MMPOG oder MMOG). Der Begriff Multiplayer bedeutet, dass an dem Spiel nicht nur eine, sondern mehrere Personen teilnehmen können. Dieser Aspekt ist auch ein besonderer Anreiz an Fußballmanagern im Web. Während Fußballmanager großer Spielefirmen wie EA⁷ potenzielle Kunden meist mit großem Funktionsumfang und graphischer Darstellung der Spiele locken, ködern die Browserbasierten Pendant ihre User im wesentlichen mit der Aussicht gegen andere reale Personen antreten zu können.

Mit Browsergame oder Browserbasiert ist gemeint, dass man um an dem Fußballmanagerspiel teilnehmen zu können nichts anderes benötigt, als einen Internetbrowser⁴ wie etwa den Microsoft InternetExplorer oder Mozilla Firefox. Denn im Gegensatz zu den Offline Managern, bei denen der Benutzer das Spiel

⁷ eine populäre kanadische Firma die Computerspiele entwickelt

installieren muss und danach die CD benötigt, läuft das gesamte Spiel im Internetbrowser ab. Die Navigation des Spiels funktioniert nur über den Browser und der Spieler hat die Möglichkeit alle Daten über den Browser abzurufen und gegebenenfalls zu verändern. Er muss also um mitspielen zu können nichts installieren.

6.3 Was das Spiel alles können soll

Wie bereits in Kapitel 2.1 erwähnt, versucht ein Fußballmanager meistens eine möglichst realistische Simulation des Alltags eines Managers eines Fußballvereins verbunden mit der größtmöglichen Portion an Spielspass zu liefern. Leider wird man jedoch im Verlauf der Programmierung bemerken, dass viele realistische Elemente den Spielspass erheblich beeinträchtigen könnten. Es gilt hier daher eine ausgewogene Mischung aus beidem zu finden.

Die Grundaufgabe eines Fußballmanagers besteht darin, Fußballspiele zu simulieren. Das heißt die Ergebnisse der Spiele zu berechnen. Diese Aufgabe wird vom Auswertungsprogramm (AWP) geregelt, welches das Herzstück des Fußballmanagers darstellt und die meiste Programmierarbeit in Anspruch nimmt. Die größte Herausforderung aber auch Anforderung an das Skript ist, dass es die Komplexität eines Fußballspiels zumindest einigermaßen nachempfinden kann. Außerdem gibt es noch eine Reihe von Rahmenaufgaben rund um das Spiel: Berechnung der Zuschauereinnahmen, der Werbeeinnahmen, der Gehaltsausgaben, noch einige andere Finanzberechnungen und die Evaluierung des Trainings. All diese Aufgaben sind allerdings ausschließlich Berechnungen im Hintergrund von denen der Benutzer nur indirekt (er sieht das Ergebnis und die Veränderungen) etwas mitbekommt. Es bleibt also noch die Anforderung den User in das Spiel zu integrieren. Also ihm den Zugang zu den Daten zu ermöglichen, und ihm einige Werkzeuge in die Hand zu legen wie er in das Geschehen eingreifen kann. Dazu gehören unter anderem, die Möglichkeiten Transfers mit anderen Vereinen zu tätigen, Taktikeinstellungen vorzunehmen, das Stadion auszubauen, Eintrittspreise festzulegen, Werbeverträge abzuschließen, Personal anzustellen und ein Jugendteam zu betreuen.

Die Simulation der einzelnen Spiele soll natürlich nicht willkürlich und einfach per Zufall stattfinden. Es muss also bestimmte Dinge geben, die anzeigen, wie gut ein Team ist und das soll nicht nur Taktik sein. Und wie gut ein Team ist, soll man wie im echten Fußball hauptsächlich daran erkennen, wie gut die Spieler sind, die in diesem Team spielen. Die Fußballmanager-Software braucht also eine Abstufung bei den Stärken der Spieler und das geschieht über die Skills (Eigenschaften). Jeder Spieler erhält nämlich mehrere Eigenschaften, die sein Können beschreiben. Diese sind beispielsweise Kopfball, Schusskraft, Passspiel, Zweikampf oder Erfahrung.

6.4 Das Ziel bei einem Fußballmanager

Jeder Spieler, der bei einem Fußballmanager antritt, hat ein Ziel: seinen Verein so gut wie möglich zu machen. Das kann man allerdings nur dann erreichen, wenn man sich als guter Manager erweist, mit seinem Budget gut haushaltet und am Ende bessere Spieler verpflichtet. Die meisten Offline-Fußballmanager sind so aufgebaut, dass der User den Computergegnern überbezogen wird, damit er eine bessere Chance nach oben zu kommen. Bei Onlinespielen hingegen haben viele User hingegen das gleiche Ziel und keiner wird bevorzugt. Das ist auf jeden Fall ein zusätzlicher Anreiz solche Spiele online zu spielen.

7 Planung und Realisierung des Fußballmanagers

7.1 Die Planung – ein entscheidender Faktor

Wenn man gleich an ein Projekt geht, ohne darüber nachzudenken, wird es wohl in den meisten Fällen scheitern. Das gilt umso mehr, wenn es sich um ein so umfangreiches Projekt wie einen Fußballmanager handelt. Es Bedarf also einer umfangreichen Planung wie man das Projekt angeht. Um dieses Problem zu bewältigen, wurden einige Modelle speziell in Bezug auf die Softwareentwicklung entworfen, die dabei helfen sollen, dass man am Ende zum gewünschten Ergebnis kommt. Da es davon aber bereits eine sehr große Anzahl gibt bräuchte man wahrscheinlich mehrere Bücher, um alle in ihren Einzelheiten zu beschreiben. Trotzdem seien zumindest zwei hier kurz angeschnitten.

7.2 Das Wasserfallmodell

Das Wasserfallmodell bezeichnet eine Vorgehensweise bei der Softwareentwicklung und wurde ursprünglich 1970 von Winston Royce entworfen. Das Modell gliedert die Entwicklung in fünf Phasen. Jede Phase hat ihren Anfangs und Endpunkt mit eindeutig definierten Ergebnissen. Am Ende steht immer die Meilensteinsitzung, bei der die Ergebnisse der vorhergehenden Phase evaluiert werden. Diese Ergebnisse gelten wie bei einem Wasserfall als bindende Vorgaben für den nächst tiefer liegenden Abschnitt.

Die Phasen des Wasserfallmodells

1. **requirement analysis and specification (Anforderungsanalyse und Spezifikation)**

Es wird eine ganz genaue Analyse erstellt, die die Anforderungen an das Endprodukt ermittelt.

2. **system design and specification (Systemdesign und Spezifikation)**

Erstellung eines Entwurfes wie das System aufgebaut sein soll und wie es funktionieren soll.

3. **coding and module testing (Programmierung und Modultests)**

In dieser Phase werden die einzelnen Teilbereiche (Module) des Endprodukts programmiert und getestet.

4. **integration and system testing (Integration und Systemtest)**

Die Module werden in das Gesamtprodukt integriert und dieses danach getestet.

5. **delivery and maintenance (Auslieferung, Einsatz und Wartung)**

Das fertige Produkt wird zum Einsatz gebracht.

Wenn man strikt nach dem Wasserfallmodell vorgeht, entsteht der Nachteil, dass alle Details bereits zu Projektbeginn feststehen müssen. Um dieser Anforderung gerecht zu werden, wird meistens ein unverhältnismäßig hoher Anteil des Aufwands in die Planungs- und Konzeptionsphase gesteckt. Zudem ist das Aufnehmen von Änderungen während des Projekts nur sehr eingeschränkt möglich. Um diese Nachteile zu beseitigen wurde das Wasserfallmodell mit Rücksprung entwickelt (Kaskadenmodell). Es erlaubt schrittweises Aufwärtsspringen, um eventuell auftretende Probleme in einer höher liegenden Phase behandeln zu können.

7.3 Prototyping

Im klassischen Sinne ist ein Prototyp die erste, bereits teilweise oder voll funktionsfähige Version, die dem angestrebten Endprodukt ähnlich ist. Das von diesem Wort abgeleitete Konzept zur Softwareentwicklung nennt sich Prototyping. Dabei werden laufend neue Versionen (Prototypen) mit immer steigendem Funktionsumfang entwickelt. Und das soll möglichst schnell und billig geschehen, wobei die Prototypen leicht modifizier- und erweiterbar sein sollten. Diese können dann relativ schnell in Bezug auf ihre Eignung eingeschätzt werden und daher weggeworfen oder weiterentwickelt werden.

Wenn es zur Entwicklung von Prototypen kommt unterscheidet man zwischen zwei Arten, vertikales und horizontales Prototyping. Beim vertikalen Ansatz werden einige der angestrebten Endfunktionen bis ins Detail simuliert, während andere vernachlässigt werden. Dadurch kann es bereits zu einer relativ realistischen Evaluation kommen, was sich jedoch auf einige kleine Teilbereiche beschränkt. Beim horizontalen Prototyping hingegen wird versucht einen möglichst breiten Überblick über das Endprodukt zu geben, wobei auf Details weitgehend verzichtet wird.

Im Zusammenhang mit Spieleprogrammierung, egal ob browserbasiert oder nicht, kommt es oftmals zur Entwicklung so genannter Alpha- und später Beta-Versionen. Das sind noch nicht vollgeständig ausgereifte aber meistens zumindest größtenteils funktionsfähige Prototypen des Endprodukts. Sie werden ausgewählten Testpersonen während eines provisorischen Testbetriebs zugänglich gemacht, um möglichst früh ein Feedback der Benutzer erhalten zu können. Die Alpha- und Beta-Versionen können demnach gezielt nach den Wünschen und Bedürfnissen der User weiterentwickelt werden. Außerdem können eventuell auftretende Probleme bereits vor dem offiziellen Spielstart beseitigt werden.

7.4 Das Spiralmodell

Eine weitere Weiterentwicklung des Wasserfallmodells ist das Spiralmodell. Es wurde 1988 von Barry W. Boehm entworfen. Das Modell sieht dabei eine zyklische Wiederholung der einzelnen Abschnitte des Wasserfallmodells vor. Denn jeder Zyklus der Spirale durchläuft einmal alle Phasen des Wasserfallmodells. Dabei nähert sich das Projekt langsam den Zielen. Am Ende jeder Windung der Spirale steht das Betrachten des Projektfortschritts (engl. review). Der entscheidende Vorteil gegenüber dem Wasserfallmodell ist die Flexibilität. So können neue Anforderungen und Ziele in das Projekt aufgenommen werden, indem man diese in der Planungsphase für den nächsten Zyklus aufnimmt. Ein besonderer Vorteil ist laut Barry W. Boehm, dass in einer Firmenstruktur das Management immer wieder eingreifen kann, ohne dabei gleich ganz den Entwicklungsprozess zu zerstören.

Beispiel für ein Spiralmodell⁸

⁸ Grafik: de.wikipedia.org



Ursprünglich war das Spiralmodell als Weiterentwicklung des bestehenden Wasserfallmodells gedacht, allerdings hat sich gezeigt, dass man viele Softwareentwicklungsmodelle durch das Spiralmodell erweitern kann oder das Spiralmodell selber mit einigen Elementen anderer Modelle. Es ist sogar möglich für jeden Spiralumlauf ein anderes Entwicklungskonzept zu verwenden. Im oben gezeigten Beispiel wird am Ende jedes Spiralzyklus ein Prototyp entwickelt - solange bis dieser betriebsfähig ist. Es ist also in gewisser Weise eine Vermischung von Spiralmodell und Prototyping.

7.5 Das von mir angewandte Modell

Für die Entwicklung meines Fußballmanagers entschied ich mich zu der gerade beschriebenen Variante, das Spiralmodell mit dem Prototyping zu kombinieren. Auf der einen Seite ist es wichtig, eine genaue Anforderungsanalyse zu machen, um bereits im Vorhinein die wichtigsten Eckpunkte des Systems zu kennen. Auf der anderen Seite ist es bei Spielen, wie bereits in Kapitel 4.3 erwähnt, oft ein Vorteil, das Spiel mit einem Prototypen (Alpha- und Beta-Version) für einige Testpersonen zu starten, da man aus den Feedbacks der User wertvolle Erkenntnisse in Bezug auf die Weiterentwicklung des Spiels gewinnen kann.

8 Effektives arbeiten mit der Template Engine

8.1 Was ist eine Template Engine

Das Wort Template ist Englisch und bedeutet übersetzt etwa Schablone. Das kommt daher, weil ein so genanntes Template ein Dokument ist, im Web meistens eine HTML-Seite, die statt mit Inhalt mit Platzhaltern gefüllt ist. Diese Platzhalter werden später von der Template Engine durch Inhalte ersetzt. Am besten kann man sich das ganze wie ein Formular vorstellen. Die Felder des Formulars (Platzhalter) werden durch die *Template Engine* mit Daten (Inhalten) gefüllt. Solche Template Systeme gibt es für fast alle bekannten Programmiersprachen.

Wenn man eine Template Engine einsetzt, benötigt man anstatt wie gewohnt eine Datei, immer zwei. Eine die die Vorlage mit den Platzhaltern enthält, und eine, die diese Platzhalter mit Inhalten ersetzt. In PHP arbeiten die meisten (wenn nicht alle) Template-Systeme mit OOP⁹.

Beispiel 1: so könnte ein HTML-Template aussehen

```
<html>
<head>
<title>HTML Template</title>
</head>
<body>
  <h1>{UEBERSCHRIFT}</h1>
  <p>{TEXT}</p><br /><br />
  <a href="javascript:history.back()">{LINK}</a>
</body>
</html>
```

Beispiel 2: ein PHP-Dokument füllt das HTML-Template mit Inhalt

```
<?php
include('./template.php');

$tpl = new Template('./');
$tpl->set_filenames(array('test' => 'htmltemplate.html'));
$tpl->setVariable('UEBERSCHRIFT', 'Html Template Test');
$tpl->setVariable('TEXT', 'Das ist ein Probetext');
$tpl->setVariable('LINK', 'zurück');
```

⁹ Objekt Orientierte Programmierung. Mehrere Daten werden in einem Objekt zusammengefasst.

```
$tpl->parse('test');  
?>
```

Beispiel 3: Ausgabe des PHP Skripts aus Beispiel 2 unter Verwendung des Templates aus Beispiel 1

```
<html>  
<head>  
<title>HTML Template</title>  
</head>  
<body>  
    <h1>Html Template Test</h1>  
    <p>Das ist ein Probetext</p><br /><br />  
    <a href="javascript:history.back()">zurück</a>  
</body>  
</html>
```

Beispiel 1 zeigt ein HTML-Template. Eigentlich ist es eine ganz normale HTML Seite mit dem einzigen Unterschied, dass der Inhalt durch Platzhalter (in geschwungenen Klammern) ersetzt ist. In Beispiel 2 ist die dazugehörige PHP-Datei. Zuerst wird mit dem Befehl `include()` die Datei in der sich die Template Engine (in dem Fall ein Objekt) befindet aufgerufen. Danach wird durch `$tpl` eine Referenz auf das Objekt erstellt. Anschließend werden dem Objekt der Dateiname der Template-Datei und drei Variablen übergeben. Am Schluss werden die Platzhalter durch die Aufruf von `$tpl->parse(,test')` durch Inhalt ersetzt und eine HTML-Datei erzeugt die wie in Beispiel 3 dargestellt aussieht.

8.2 Wieso eine Template Engine verwenden?

Programmieren mit *Template Engine* hat viele entscheidende Vorteile. Zum einen wird der Programm-Code (in meinem Fall PHP) strikt vom Design (HTML) getrennt. Dadurch wird der Effekt erzielt, dass sich sowohl Code als auch Design um ein vielfaches leichter warten lassen. Des Weiteren werden Webprojekte um ein vielfaches schneller entwickelt, wenn man sich erst einmal an die Template Engine gewöhnt hat. Wenn mehrere Personen an einem Webprojekt beteiligt sind, entsteht noch ein weiterer großer Vorteil: Programmierer und Designer können unabhängig voneinander arbeiten.

Einzigster Nachteil einer Template Engine ist, dass ein so genannter Overhead entsteht. Das bedeutet, dass mehr Dateien zu speichern sind als eigentlich benötigt werden, da man statt nur einer PHP Datei pro Seite noch eine zweite Template Datei benötigt. Da aber in der heutigen Zeit Webspaces auch in großen Mengen sehr billig geworden ist stört dieser Nachteil nicht im Geringsten.

9 Objektorientierte Programmierung (OOP)

9.1 OOP ganz allgemein

OOP ist die Kurzform für Objektorientiertes Programmieren und bezeichnet ein Konzept der modernen Informatik, das im Gegensatz zur klassischen Prozeduralen Programmierung, die strikt zwischen Daten und Prozeduren unterscheidet, beides als untrennbare Teile des Ganzen oder als Einheiten betrachtet. Diese Einheiten nennt man Objekte (deswegen auch Objektorientierte Programmierung). Bei dem Verfahren der Objektorientierten Programmierung werden Daten zu Attributen (Eigenschaften) des Objekts und Prozeduren zu Methoden.

„Zumindest konzeptionell arbeitet ein Programm dann [wenn es objektorientiert programmiert ist] nicht mehr [wie bei der prozeduralen Programmierung] so, dass sequenziell einzelne Funktionsbereiche eines Algorithmus durchlaufen werden, die dabei eine Anzahl Daten verändern, sondern die Programmlogik entfaltet sich in der Kommunikation und den internen Zustandsveränderungen der Objekte, aus denen das Programm aufgebaut ist.“

(Zitat aus Wikipedia¹⁰)

9.2 Wieso OOP verwenden?

Ein Vorteil der Objektorientierten Programmierung ist, dass durch den Zusammenschluss zusammengehöriger Elemente und Methoden eine bessere Modularisierung des Codes erreicht wird. Damit verbunden sind auch bessere Wartbarkeit und die Möglichkeit der Wiederverwendung der einzelnen Module sowie die höhere Programmflexibilität, insbesondere in Bezug auf die Benutzerführung.

Objektorientierte Programmierung kommt bei meinem Fußballmanager sehr oft zum Einsatz. Erstens wird der Datenbankzugriff durch ein Objekt gehandhabt. Des Weiteren greift auch die Auswertung der Einzelnen Spiele auf die Technik der objektorientierung zurück.

¹⁰ Zitat: http://de.wikipedia.org/wiki/Objektorientierte_Programmierung, 15.1.2006

9.3 Grundlegende Prinzipien von OOP

Die Konzeption von Objektorientierter Programmierung erfolgt in der Regel nach bestimmten Programmierparadigmen. Die wichtigsten davon möchte ich an dieser Stelle kurz vorstellen, da dies zum besseren Verständnis von Objektorientierter Programmierung verhilft.

Abstraktion

Abstraktion beschreibt die Reduktion einer Klasse auf das Mindestmaß.

Kapselung

Jedes Objekt hat eine Schnittstelle über die man mit dem Objekt kommunizieren kann. Diese Schnittstelle, bestimmt in wie weit mit dem Objekt interagiert werden kann.

Polymorphie

Durch Polymorphie wird ermöglicht, dass bestimmte Prozeduren auf unterschiedlich getypte Parameter¹¹ unterschiedlich reagieren.

Vererbung

Vererbung erlaubt es neue Objekte zu definieren auf Basis bereits bestehender. Dabei werden die Eigenschaften und Methoden des Basisobjekts übernommen und können mit weiteren aufgefüllt oder überlagert werden.

¹¹ Werte die einer Funktion (bei OOP Methode) übergeben werden

10 Cronjobs

10.1 Das Spiel läuft von selbst

Der Spielablauf funktioniert, sofern nicht anders festgelegt, automatisch. Es bedarf niemanden der einen Knopf drückt oder einen Hyperlink anklickt, damit irgendeine Aktion des Spiels ausgelöst oder gestartet wird. Die Auswertung der Spiele oder des Trainings passiert zu einer festgelegten Zeit von ganz alleine. Denn die Auslöser solcher Aktionen sind so genannte Cronjobs, sie „steuern“ das Spiel.

10.2 Was ist ein Cronjob?

Als Daemon (**d**isk and **e**xecution **m**onitor, Schreibweise auch häufig Demon) bezeichnet man ein Programm unter UNIX oder Unix-artigen Betriebssystemen, welches im Hintergrund läuft und bestimmte Dienste zur Verfügung stellt. Einer dieser Dienste ist der Cron-Daemon. Der Cron-Daemon steuert wiederkehrende Aufgaben, die Cronjobs, welche zu einer bestimmten Zeit ausgeführt werden sollen. Diese Cronjobs werden in einer Crontab gespeichert, die die Angaben zur gewünschten Zeit und dem Befehl enthält. Die Crontab besteht aus 6 Spalten, wobei 5 der Zeitangabe und eine der Angabe des Befehls dienen. Es besteht die Möglichkeit als Zeitangabe auch jede Stunde, jeden Tag, jede Woche oder jedes Monat anzugeben.

10.3 Die Crontab für den Fussballmanager

Für meinen Fußballmanager werden insgesamt drei Cronjobs benötigt. Meine Crontab sieht daher folgendermaßen aus:

```
#M S T W M Befehl
#-----
0 0 * * * wget [Pfad]/training.php > /dev/null
0 22 * * * wget [Pfad]/auswertung.php > /dev/null
0 6 * * * wget [Pfad]/usercontrol.php > dev/null
#-----
```

Die erste Aufgabe ruft jeden Tag um Punkt Mitternacht das Skript *training.php* auf welches das Training berechnet. Die zweite setzt, indem sie das Skript

auswertung.php aufruft jeden Tag um 22:00 Uhr die Auswertung der an diesem Tag anstehenden Spiele. Der letzte Cronjob startet das Skript, welches überprüft, ob die User noch aktiv sind und löscht gegebenenfalls überflüssige Benutzer aus der Datenbank.

11 Das Herzstück des Spiels – Die Spielsimulation

11.1 Die Aufgaben des AWP

Wie bereits in Kapitel 3.3 erwähnt ist die Grundaufgabe des Spiels die virtuellen Fußballspiele zu simulieren, das mittels Auswertungsprogramm (AWP) geschieht. Hierbei kommen dem AWP für jedes Spiel drei Blöcke an Aufgaben zu.

1. Berechnungen vor dem Spiel

Bevor das Spiel beginnen kann, müssen zuerst einmal alle für das Spiel beziehungsweise für das Umfeld relevanten Daten ermittelt werden. Dazu gehören unter anderem der Zuschauerzustrom sowie die daraus resultierenden Zuschauereinnahmen für den Heimverein, das Wetter, das leichten Einfluss auf den Punkt Zuschauerzustrom und das Spiel hat, die von den Teams eingestellten Aufstellungen und Taktiken sowie der Bewerb in dem das Spiel stattfindet.

2. Die Eigentliche Simulation des Spiels

Ein Fußballspiel zu simulieren, bedeutet nichts anderes als irgendwie zu einem Ergebnis zu kommen. Das könnte zum Beispiel eine einfache Formel sein. Allerdings ist ein Fußballmanager, welcher nach diesem Verfahren vorgeht und die Ergebnisse einfach nur berechnet und anzeigt auf Grund der großen Konkurrenz für die meisten potenziellen Spieler uninteressant. Es müssen schon mindestens die Torschützen ausgegeben werden. Jedoch wollte ich in meinem Skript noch eine weitaus größere Herausforderung implementieren, eine Textauswertung. Von einer Textauswertung kann man dann sprechen, wenn der User das Ergebnis nicht nur in Zahlenform (zum Beispiel „0:3“) geliefert bekommt, sondern auch einen Text sieht der den Spielverlauf oder zumindest die Höhepunkte dokumentiert.

3. Berechnungen nach dem Spiel

Genauso wie vor dem Spiel kommen auch nach dem Spiel noch einige Aufgaben auf das Skript zu. Zuerst muss überprüft werden ob es zu einer Verlängerung kommen muss und wenn das der Fall ist muss das Skript noch einmal in die Simulation zurückspringen. Danach müssen die Daten des Spiels verarbeitet werden, also das Ergebnis und die Textauswertung gespeichert werden. Das gleiche geschieht mit den Daten der Spieler. Hier ist es notwendig eventuelle Sperrern für die nächsten Spiele zu evaluieren und alle Statistiken der Spieler für das Spiel zu speichern.

Da sich die wirkliche Spielsimulation nur und Punkt zwei abspielt, und die anderen Punkte im Vergleich relativ uninteressant und unspektakulär sind wird sich das folgende Kapitel im Allgemeinen mit diesem Punkt befassen.

11.2 Das Grundprinzip

Zu Grunde liegt dem Auswertungsprogramm eigentlich ein ganz simples Prinzip: Jedes Team stellt seine Aufstellung und seine Taktik. Die Aufstellung besteht, wie beim echten Fußball natürlich, aus elf Spielern. Jeder dieser elf Spieler hat eine gewisse Stärke, die sich aus einigen Einzelstärken verschiedener Eigenschaften zusammensetzt (Vgl. Kapitel 3.3). Die Taktik besteht aus einigen Einstellungen, wie zum Beispiel Einsatz oder Spielweise. Hauptsächlich aus diesen Komponenten setzt sich die Berechnung des Spiels zusammen.

Gearbeitet wird mit einem Objekt. Dieses Objekt enthält einige Variablen (Eigenschaften), die den Zustand des Spiels beschreiben und einige Funktionen (Methoden), die ihn verändern. Die Eigenschaften, die das Spiel beschreiben, beinhalten Informationen zu den Zuschauern, zum Wetter, zu den Teams und den Spielern, zu den taktischen Einstellungen der Teams und der momentanen Lage des Spiels.

11.3 Die eigentliche Simulation des Spiels

Wenn die Berechnungen aus Block 1 abgeschlossen sind, gelangt man zum Herzen des AWP, das aus einer Schleife besteht. Diese Schleife wird für jede Minute des Spiels einmal durchlaufen. Im Inneren der Schleife laufen einige Berechnungen ab, die feststellen, ob einem Team ein Angriff gelingt. Wenn das der Fall ist, wird eine entsprechende Funktion gestartet. Faktoren für diese Berechnungen sind die Taktiken der Teams und die Spielerstärken.

Jede Aktion des Spiels wird durch eine Funktion beschrieben, die den Ablauf und Ausgang berechnet. Die Berechnung funktioniert dabei im Wesentlichen so: Ein

Spieler soll eine Aktion (zum Beispiel einen Pass oder Schuss) ausführen. Dazu wird die betreffende Eigenschaft „auf die Probe gestellt“, es wird die Funktion `probe()` aufgerufen. Diese Funktion berechnet einen Rückgabewert. Dieser wird auf Basis von Wahrscheinlichkeiten ermittelt. Der Rückgabewert dieser Funktion ist ein Indikator wie gut der Spieler die Probe geschafft hat und liegt in einem Wertebereich von 0 (sehr schlecht) – 6 (sehr gut). Die Funktion die die Aktion berechnet arbeitet auf Basis dieses Ergebnisses weiter. So wird zum Beispiel wenn der Wert 0 ist, also die Probe schlecht ausgegangen ist, ein Fehlpass ausgeführt und die Funktion verlassen. Wenn der Wert aber 6 ist, so glückt dem Spieler ein Pass und das Ziel wird erreicht, ein Mannschaftskollege des Spielers kann eine weitere Aktion ausführen.

11.4 Die Probe

Nehmen wir die Funktion `probe()` einmal etwas genauer unter die Lupe.

Beispiel 4: die Probe-Funktion

```
<?php
function probe($id,$eigen,$plus = 0) {

    ##### SPIELER EIGENSCHAFT
    $eig = $this->spieler_eig[$id][ "$eigen" ];
    ##### FIT, MOR, FORM, KON, SEL
    $a1 = ($this->spieler_eig[$id][ "FIT" ]+15)/100;
    $a1 = min(1,$a1);
    $a2 = ($this->spieler_eig[$id][ "MOR" ]+10)/100;
    $a2 = min(1,$a2);
    $a3 = (($this->spieler_eig[$id][ "FORM" ]+3)/10);
    $a3 = max(0.6,$a3);
    $a4 = (($this->spieler_eig[$id][ "SEL" ]+20)/100);
    $a4 = max(0.6,$a4);

    ##### ERFAHRUNGS FAKTOR
    $b1 = 1; $erf = $this->erfart;
    $b1 = (($this->spieler_eig[$id][ $erf ]+1)/500)+0.8;

    ##### WETTER FAKTOR
    $wetter_faktor = 1;
```

```
    if($this->wetter == 1 and ($eigen == "SCP" or $eigen ==
"GES" or $eigen = "BAK" or $eigen = "TEC")) {
        $wetter_faktor -= 0.1;
    } // if
    if($this->wetter == 2 and ($eigen == "BAK" or $eigen ==
"TEC" or $eigen == "ZUV")) {
        $wetter_faktor -= 0.1;
    } // if
    if($this->wetter == 3 and ($eigen == "BAK" or $eigen ==
"TEC" or $eigen == "ZUV")) {
        $wetter_faktor -= 0.05;
    } // if

    ##### ZUSAMMENSETZEN DES FAKTORS
    $faktor = (($a1*2) + ($a2) + (2*$a3) + (2*$a4) + $b1 +
$a5)/9;

    ##### SPIELER END EIGENSCHAFT
    $task = $eig*$faktor*$wetter_faktor;

    ##### RANDOM ZAHL
    $maxim = ($this->lz*1.4);
    $frand = rand(0, $maxim+$plus);

    ##### RÜCKGABEWERT ERMITTELN
    $finalTask = $task - $frand;
    switch(true)
    {
        case $finalTask > $eig * 0.95:
            $level = 6;
            break;
        case $finalTask > $eig * 0.85:
            $level = 5;
            break;
        case $finalTask > $eig * 0.65:
            $level = 4;
            break;
        case $finalTask > $eig * 0.45:
            $level = 3;
            break;
        case $finalTask > $eig * 0.30:
            $level = 2;
```

```
        break;
    case $finalTask > $eig * 0.20:
        $level = 1;
        break;
    default:
        $level = 0;
} //switch

#### SPIELER DONE
$this->spieler_stat["donelevel"][$sid] += $level;
$this->spieler_stat["done"][$sid]++;

#### RETURN
return $level;

} /// function
?>
```

Was macht diese Funktion jetzt? Zu allererst müssen der Funktion mindestens 2 Parameter übergeben werden, die ID des Spielers (eine Identifikationsnummer) und die Eigenschaft die getestet werden soll. Dann werden der Wert dieser Eigenschaft und einige Faktoren, die die Eigenschaft verändern berechnet, die Fitness, die Moral, das Selbstvertrauen, die Form, die Erfahrung sowie das Wetter. Danach wird der Wert der Eigenschaft mit dem zusammengesetzten Faktor multipliziert und eine Zufallszahl ermittelt und diese dann von der Eigenschaft abgezogen. Nun wird je nach dem wie viel Prozent der Anfangsstärke der Eigenschaft dieser Wert ergibt der Rückgabewert zwischen 6 (seht gut) und 0 (sehr schlecht) ermittelt. Dieser wird dann noch gespeichert und zurückgegeben. Die Funktion, die nach einer Probe verlangt hat, kann jetzt auf Basis dieses Wertes weitere Berechnungen durchführen.

12 Anhang

12.1 Anhang A - Quellenverzeichnis

- Krause, Jörg: Programmieren lernen in PHP4. Ein kompakter Einstieg in die Webserverprogrammierung. 2.Auflage – München Wien: Carl Hanser Verlag, 2003
- Günther, Ulrich: PHP – Ein praktischer Einstieg. 1.Auflage – Köln: O'Reilly Verlag GmbH & Co. KG, 2003
- Bergmann, Sebastian: Professionelle Softwareentwicklung mit PHP 5. Objektorientierung, Entwurfsmuster, Modellierung und fortgeschrittene Datenbankprogrammierung. 1.Auflage – Heidelberg: dpunkt.verlag, 2005
- Hanke, Johann-Christian: PHP für Einsteiger. 1.Auflage – Osnabrück: KnowWare, 2002-07
- <http://de.wikipedia.org>
- <http://www.php.net>
- <http://www.php3.de>
- <http://www.php.de>
- <http://www.schattenbaum.net>
- <http://www.selfphp.de>
- <http://www.selfhtml.de>
- <http://www.pc-magazin.de>
- <http://www.tu-chemnitz.de/docs/mysql/>
- <http://en.wikipedia.org>
- <http://www.mysql.com>
- http://www.heise.de/ct/schan/server_dok_31/kap68.htm
- <http://www.plmlabor.de/index.php?id=1443>
- <http://apfelwiki.de/wiki/Main/CRON>
- <http://www.usability-forum.com/bereiche/Prototyping.pdf>
- <http://www.projektmagazin.de/glossar/gl-0825.html>
- <http://www.selfphp.de/praxisbuch/index.php>
- http://www-user.tu-chemnitz.de/~dors/modelle_wasser.html
- <http://www.projektmagazin.de/glossar/gl-0824.html>

- <http://www.plmlabor.de/index.php?id=1435>
- http://www-user.tu-chemnitz.de/~dors/modelle_spiral.html
- <http://www.projektmagazin.de/glossar/gl-0099.html>
- <http://www.ralfbuerger.de/sse/Vorgehensweise/Modelle.htm>

12.2 Anhang B – Links zu webbasierten Projekten

Der Fußballmanager im Web: <http://www.psw-manager.de>

Ein weiteres Browserbasiertes MMOG von mir: <http://www.tennispur.com>

12.3 Anhang C – Quellcodeverzeichnis

Beispiel 1: So könnte ein HTML-Template aussehen	- 19 -
Beispiel 2: Ein PHP-Document füllt das HTML-Template mit Inhalt	- 19 -
Beispiel 3: Ausgabe des PHP Skripts aus Beispiel 1 unter Verwendung des Templates aus Beispiel 2.....	- 20 -
Beispiel 4: die Probe-Funktion	- 28 -

12.4 Anhang D – Zur CD

Beiliegend zu meiner Fachbereichsarbeit finden sie eine CD. Auf dieser CD finden sie eine Demo der Spielauswertung, die Quellcodes der Template Engine und der Spielauswertung sowie eine Word-Version der Fachbereichsarbeit.

Um die CD benutzen zu können wird ein installierter Virtual Webserver, der PHP und MySQL unterstützt, benötigt. Sollten sie noch keinen Virtual Webserver installiert

haben können sie unter www.apachefriends.org einen herunterladen. Die Installation ist einfach und auf der Website befindet sich eine Anleitung.

Wenn ein Virtual Webserver installiert ist, dann öffnen sie die CD und kopieren den Ordner „fba“ in das Hauptverzeichnis ihres Virtual Webserver. Rufen sie in ihrem Browser die Adresse „<http://localhost/fba>“ auf und befolgen sie die Installationanweisungen. Danach stehen ihnen alle Features der CD zur Verfügung.

12.5 Anhang E – Danksagung

Ich möchte mich Ende noch bei all jenen Bedanken die mir im Laufe der Entwicklung meiner Software und der Fachbereichsarbeit geholfen haben.

- Den Usern www.php.de, die mir immer in verzwickten lagen mit ihrem Rat zur Seite standen.
- Den Mitwirkenden bei de.wikipedia.org, die mit wissenswerten Beiträgen und Linksammlungen die Recherche für die Fachbereichsarbeit etwas erleichterten.
- www.schattenbaum.net, dafür, dass mir bei einem Einstieg in PHP geholfen wurde.
- Meiner Familie, die mich in Ruhe arbeiten gelassen hat.
- Und natürlich auch meinem Betreuer Carl Metnitz, der in allen Fragen rund um meine Fachbereichsarbeit ein kompetenter Berater war.