

**Fachhochschul-Diplomstudiengang für  
Telekommunikation und Medien**

**MMOFG –  
Massive Multiplayer Online Flash Games**

Wien, im August 2006

Diplomarbeit

„MMOFG – Massive Multiplayer Online Flash Games“

Ausgeführt zum Zweck der Erlangung des akademischen Grades eines

**Dipl.-Ing. (FH) Telekommunikation und Medien**

am Fachhochschul-Diplomstudiengang Telekommunikation und Medien St. Pölten

unter der Erstbetreuung von

Markus Seidl Bakk.

Zweitbegutachtung von

Dipl.-Inform. Joachim Baumert

ausgeführt von

Andreas Winkler  
tm021125

St. Pölten, am 4. September 2006

Unterschrift:

## Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Diplomarbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.
- ich dieses Diplomarbeitsthema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der von den Begutachtern beurteilten Arbeit überein.

..... Ort, Datum

## Kurzfassung

Moderne Computerspiele sind bereits lange dem Garagenzeitalter entwachsen. Teams aus mehreren Dutzend Programmierern, Grafikern und Testern entwickeln teils in mehrjähriger Arbeit Spiele für mehrere Millionen Euro. Besonders im Bereich der Massively Multiplayer Onlinespiele sind Aufwand und Kosten für ein erfolgreiches Produkt extrem hoch. Andererseits bietet genau dieser Markt Platz für Spiele, deren Entwicklung wieder an die Zeit erinnert, bevor internationale Publisher die Branche übernommen haben. So genannte Browserspiele werden heute noch von einzelnen Entwicklern mit überschaubarem Aufwand erstellt und brauchen sich teilweise bezüglich Spieltiefe und Communitygröße nicht vor kommerziellen Produkten zu verstecken.

Die vorliegende Arbeit analysiert den Versuch, ein eben solches Browserspiel zu entwickeln. Es soll die Möglichkeit aufgezeigt werden, mit personell wie finanziell geringem Aufwand ein eigenes Spiel zu entwickeln.

Zu Beginn werden die theoretischen Grundlagen bezüglich Onlinespiele und Virtual Societies abgehandelt, bestehend aus Geschichte, Marktübersicht und Entwicklungspotential. Außerdem wird ein Überblick über die verschiedenen Typen von Spielern in Massively Multiplayer Online Games gegeben.

Aus diesen Grundlagen wird ein Spielkonzept entwickelt, das im praktischen Teil der Arbeit umgesetzt wird. Die Beschreibung der Umsetzung und der aufgetretenen Probleme während der Entwicklung führt schließlich zu einer Reflexion, die Fehler aufzeigt, mögliche Lösungswege beschreibt und Verbesserungs- und Erweiterungsmöglichkeiten vorstellt.

## **Abstract**

Modern computer games have long outgrown the garage. Teams consisting of dozens of programmers, designers and testers need years to develop million-euro games. The costs and effort needed to create a successful product are extremely high, especially for Massively Multiplayer Online Games. On the other hand, this market makes way for games which remind us of the times before international publishers took over the market. So-called browser games are nowadays created with manageable effort by single developers and often these are in no way inferior to commercial products in terms of complexity and community size.

The attempt to create such a browser game is analysed in this thesis in order to point out the possibility for developing a computer game with few demands on human resources and time. The first part discusses the principles of online games: history, market overview and future potential for development. Furthermore, a survey of player types is given.

Based on these principles, a concept is developed which is implemented in the practical part. The description of the implementation phase including recurring problems leads on to a reflection on failures, describing possible solutions and presenting ways to improve and upgrade the game.

# Inhaltsverzeichnis

1 Einleitung.....	8
2 Onlinespiele.....	9
2.1 Geschichte .....	10
2.2 (Flash) Browserspiele .....	16
Interaktion .....	17
Genres .....	18
Marktübersicht.....	19
2.3 Aufbau von Onlinespielen .....	22
turn based .....	23
tick based .....	24
Kommunikation.....	24
2.4 Communities / Virtual Societies .....	26
2.5 Spielertypen .....	28
2.6 Designprinzipien .....	34
3 Proof of Concept.....	37
3.1 Ziele .....	38
3.2 Spielkonzept .....	38
Technischer Aufbau .....	40
Spielablauf .....	41
3.3 Umsetzung.....	45
Konzeptionsphase.....	46
Umsetzungsphase.....	47

4. Ergebnisse .....	57
4.1 Beantwortung der Fragestellung .....	57
4.2 Schwierigkeiten bei der Erstellung .....	58
4.3 Spielstatus .....	59
4.4 Erweiterungsmöglichkeiten und Ausblick .....	60
5. Schlusswort .....	61
Quellenverzeichnis .....	62
Abbildungsverzeichnis .....	64
Glossar .....	65

## 1 Einleitung

*Ein ständig wachsendes Universum, bestehend aus unzähligen Planeten, jeder unter der Kontrolle eines machtgierigen Herrschers. Kämpfe und technologische Fortschritte stehen an der Tagesordnung, Allianzen werden gegründet, Bündnisse gebrochen und interstellare Kriege gigantischen Ausmaßes ausgefochten.*

Um an einem solchen oder ähnlichen Szenario teilzunehmen, ist es heutzutage nicht nötig ein teures Computerspiel zu kaufen und seinen Rechner kostspielig auf dem neuesten Stand der Technik zu halten. Browserspiele bieten große Communities und nie da gewesene Spieltiefe zum Nulltarif. Hinter diesen Spielen stehen jedoch (noch) nicht die bekannten Publisher und große Entwicklungsstudios, sondern einzelne, passionierte Entwickler, die solche Onlinewelten in ihrer Freizeit erschaffen.

Ein Browser und Internetzugang genügen, um als mittelalterlicher König oder interstellarer Warlord Geschichte wie Zukunft zu verändern, oder als Manager eines Kreisligavereins die Champions League zu gewinnen. Für die Entwickler solcher Spiele ist es jedoch ein steiniger Weg von der ersten Idee bis zur bevölkerten Onlinewelt: heterogene Entwicklungsumgebungen müssen zusammengeführt und Ressourcen- und Performanceflaschenhalse bewältigt werden. Am Ende steht jedoch die Genugtuung, alleine eine Welt geschaffen zu haben, die Tausende Spieler begeistert.

## 2 Onlinespiele

In diesem Kapitel wird versucht, einen Überblick über die Welt der Onlinespiele zu geben. Zuerst ist zu definieren worum es sich dabei überhaupt handelt, danach folgt ein Abstecher in die Geschichte dieser Computerspiele, um die Wurzeln virtueller Realitäten freizulegen. In weiterer Folge werden Ablauf, Design und nicht zuletzt die Spieler, die ja schließlich den Kern jeder Onlinewelt ausmachen, beleuchtet.

Ein Onlinespiel definiert sich, wie der Name schon sagt, darüber, dass es nicht auf den Rechner des Spielers beschränkt ist, sondern vielmehr über das Internet gespielt wird. Der Begriff kann weit gefasst werden und beinhaltet dann auch die unzähligen Pausenfüller auf Flash-Basis, die hauptsächlich auf den großen Internetportalen von Yahoo! bis AOL angeboten werden.

Klassischer betrachtet sind mit Onlinespielen aber virtuelle Welten gemeint, die von hunderten bis tausenden Spielern bevölkert werden. Diese Massively Multiplayer Online Games (MMOGs) bieten eine von einem oder mehreren Servern verwaltete Spielwelt, in die sich jeder Spieler, meist mittels eines Clients, jederzeit einloggen und verschiedene Aktionen ausführen kann.

Weiters lassen sich Onlinespiele in Browserspiele und clientbasierte Spiele unterteilen. Während Browserspiele in HTML oder Macromedia (mittlerweile Adobe) Flash mit Hilfe von serverseitigen Scriptsprachen und Datenbanken entwickelt werden und ohne zusätzlichen Installationsbedarf im Webbrowser des Spielers laufen, muss bei clientbasierten Spielen eine spezielle Software (eben der Client) installiert werden. Die Clientsoftware ist meist plattformabhängig und funktioniert somit nur auf den Betriebssystemen für die sie entwickelt wurde. Technologisch fußen diese Spiele meist auf einer Client-Server-Architektur, es sind aber auch Implementierungen auf Basis eines Peer-to-Peer-Netzes möglich. Browserspiele hingegen sind grundsätzlich unabhängig vom Betriebssystem und funktionieren in jedem Webbrowser, der die verwendeten Technologien unterstützt. Spiele auf Flashbasis setzen zusätzlich ein Flash-Plugin, das es für die meisten Browser gibt, voraus.

Auf den Inhalt bezogen unterscheidet man folgende drei Typen von clientbasierten Onlinespielen: Classic Games, Retail Hybrid und Persistent World (PW).

Ersterer Typ beinhaltet Umsetzungen von klassischen Brettspielen wie Schach oder Mah Jong und ist hauptsächlich auf Internetportalen wie Yahoo! vertreten. Als Retail Hybrid werden Spiele bezeichnet, die zwar als Offlinespiel verkauft werden, jedoch Möglichkeiten zum Spielen über das Internet bieten. Hierzu zählen ein Großteil der aktuellen 3D-Actionspiele sowie Strategie- und Sporttitel. PWs werden beispielsweise von Spielen wie *Ultima Online*, *World of WarCraft* oder *Everquest* vertreten und sind auf das Online-Spielerlebnis reduziert – sie bieten somit keinen Offlinepart. [MULL02 S.8f]

## 2.1 Geschichte

Da alle modernen Computerspiele, somit auch Browser Spiele, ihre Wurzeln weit in der computergeschichtlichen Vergangenheit haben, lohnt sich ein Blick auf die Ursprünge dieser Form der Unterhaltung.

Im Gegensatz zur landläufigen Meinung sind Onlinespiele beinahe so alt wie Computerspiele selbst. [FRIE02 S.4] Auch in Zeiten vor der Entwicklung des Internet gab es vernetzte Rechner und damit die Möglichkeit Spiele nicht nur alleine zu genießen.

Während das allererste Computerspiel, „Pong“ von William A. Higinbotham, bereits 1958 im Zuge eines Tags der offenen Tür am Brookhaven Gymnasium einer breiteren Öffentlichkeit zugänglich gemacht wird, dauert es noch gut zehn Jahre bis zum ersten echten Multiplayer-Spiel. [http8BIT], [MERT06]

Das 1961 an der Universität von Illinois eingeführte Multi-User-System PLATO wird 1969 von Rick Blomme für *Spacewar* missbraucht. Zwar ist dies keine Neuentwicklung sondern vielmehr eine Adaption eines am MIT auf einer PDP-1 entwickelten Spiels (siehe Abb. 2.1, S. 11), kann aber auf jeden Fall als die Geburtsstunde der Netzwerkspiele bezeichnet werden. Im Gegensatz zu *Spacewar*, das nur für zwei Spieler nutzbar war, bietet die *Star Trek*-Verarbeitung

*Empire* bereits 1972 Platz für bis zu 32 Spieler, um sich im Weltraum zu messen. Die nächsten Jahre sind geprägt von der Entstehung der ersten umfangreicheren Spiele, beispielsweise *Avatar*, das Erstgeborene aus der *Wizardry*-Serie und *Oubliette*.



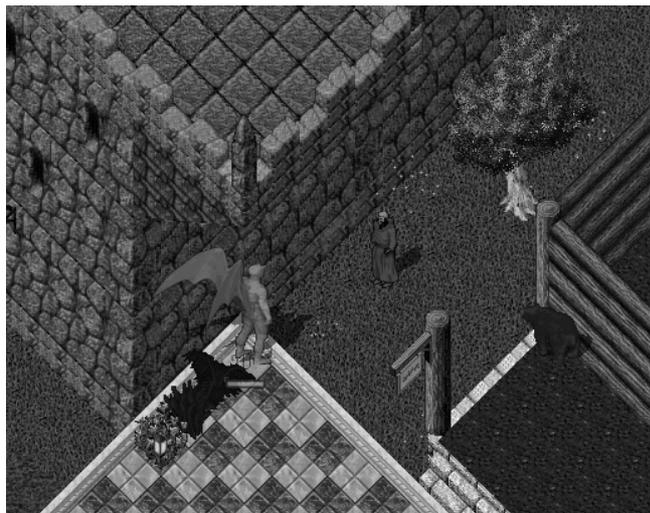
Abbildung 2.1 Nachbildung von Spacewar in Java [http:MIT]

Erwähnenswert ist auch, dass Zweiteres bereits eine rudimentäre 3D-Ansicht bot, die aus einzelnen Linien gezeichnet wurde. Dabei ist festzuhalten, dass die Benutzeroberflächen der damaligen Spiele und damit bis zu einem gewissen Grad auch das Spielerlebnis, eher spröde ausfielen. So musste man, um einen Zauberspruch zu benutzen, dessen Namen eintippen – je schneller desto besser. 1978 folgt einer der großen Meilensteine der Onlinespiele-Geschichte: *MUD1*, von Roy Trubshaw und Richard Bartle. Dieser *Multi User Dungeon* (MUD) wird an der Universität von Essex entwickelt und als 1980 der Anschluss an das ARPANet, den Vorläufer des heutigen Internets, folgt, verbreitet sich das Spiel (und sein Code) schnell um den Globus. Viele Studenten beginnen nun ihre eigenen MUDs zu basteln, verändern und verbessern die Algorithmen und realisieren neue Konzepte und Ideen. Daraus entwickeln sich viele Varianten des ursprünglichen Spiels, die auch eigene Akronyme bekommen (*MOO*, *MUSH*, *MUCK*, *MUSE* und *MAGE*).

In den frühen 1980er-Jahren beginnt der erste Internetprovider CompuServe das Potential von Multiplayer-Spielen zu entdecken und beauftragt die Kesmai Corp. damit, ein textbasiertes Online-Rollenspiel zu entwickeln. Das Ergebnis nennt sich *Islands of Kesmai* und markiert zusammen mit dem Erstlingswerk dieser Firma, *Megawars I*, den Beginn der kommerziellen MMOGs – mitsamt astronomischer Gebühren von \$12 pro Stunde.

In der zweiten Hälfte dieses Jahrzehnts versuchen weitere Firmen wie GENie und QuantumLink, dem Vorläufer von AOL, am noch jungen Markt der Onlinespiele zu partizipieren. [FRIE02 S.6f]

Die zweite Hälfte der 80er Jahre ist geprägt von den ersten kommerziellen Onlinespielen und dem Auftauchen von „echter“ Grafik in diesem Sektor (z.B. *Rabbit Jack's Casino* oder *Air Warrior* von Kesmai). Auch die ersten E-Mail-Spiele auf kommerziellen Servern erblicken mit *Rim Worlds War* von Jessica Mulligan das Licht der Welt. GENie wird in dieser Zeit zum größten Anbieter von MMOGs und lizenziert eine große Anzahl von Spielumgebungen, darunter *Galaxy II* und *Orb Wars*. Mit *A-Maze-ing* erscheint 1989 schließlich der erste Multiplayer-3D-Shooter. Obwohl die Penetration der Haushalte mit (Modem-) Internetzugängen zu dieser Zeit noch sehr schwach ist, veröffentlicht Electronic Arts mit *Modem Wars* 1988 das erste Spiel, das hauptsächlich darauf ausgelegt ist, über ein Modem gegeneinander zu spielen.



**Abbildung 2.2** Screenshot aus Ultima Online [httpUO]

Während zu Beginn der 90er Jahre alle großen Anbieter von Multiplayer-Spielen über die Online-Lizenz des Fantasy-Rollenspiels *Ultima* verhandeln, erscheint 1991 *LambdaMOO*, ein MUD-basiertes Onlinespiel, das einige soziale Aspekte von Virtual Communities aufwirft und großen Einfluss auf die weiteren Entwicklungen hat. Als erstes Spiel erlaubt es eine virtuelle Demokratie ohne den Einfluss der Spieldesigner aufzubauen und kämpft schließlich mit Problemen wie Gewalt, sexueller Belästigung und sogar einer Vergewaltigung – und das alles innerhalb einer rein textbasierten Onlinewelt. Durch diese Vorgänge müssen erstmals Aspekte wie Verantwortung der Spieler für ihre Aktionen und virtuelle Gesetze erörtert werden. 1993 ist jedoch auch aus einem anderen Grund für die Entwicklung von Multiplayer-Spielen von großer Bedeutung, da sich das ARPAnet bereits zum WWW weiterentwickelt hat und nun mit *Mosaic*, dem Vorläufer des *Netscape Navigators*, erstmals grafische Präsentation möglich wird. Durch den beginnenden Internet-Boom fallen auch die Preise für MMOGs auf einen durchschnittlichen Wert von \$3 pro Stunde. Zu dieser Zeit beginnen die Hersteller von Offlinespielen auch damit, ihre Produkte mit Netzwerk- und Internetmodi aufzuwerten. Durch die große Popularität von ID Software's *Doom*, *Doom II* und *Quake* entwickelt sich 1996 ein neuer Sektor auf dem Computerspielmarkt und Internetfähigkeiten werden zu einem Muss für Spiele. Die Verbreitung von onlinefähigen Spielen führt zur Entstehung von vielen Internetportalen die jeweils einige Spiele unterstützen und verschiedene Mechanismen anbieten, damit die Spieler in diesem entstandenen Dschungel passende Kontrahenten finden können. Bald steigen auch die Spielefirmen selbst in diesen Bereich ein und beginnen eigene Services aufzubauen – allen voran *Blizzard Entertainment* mit *Battle.net*. [LOBER02]

Neben diesem Trend entwickelt sich auch die Grafik der reinen Onlinespiele weiter und mit *Meridian 59* [httpM59] von 3DO, einer Persistent World in 3D, erscheint quasi ein Begründer des Genres der MMORGs (Massively Multiplayer Online Role Playing Games). Auf dieses noch relativ erfolglose Spiel folgen mit dem lange erwarteten *Ultima Online* [httpUO] (siehe Abb. 2.2, S. 12), das bereits 50.000 Spieler in den ersten drei Monaten für sich gewinnen kann und den, wie man in den folgenden beiden Abbildungen sehen kann, grafisch aufwändigeren

Titeln *Asheron's Call* [[httpAC](http://AC)] von Turbine Entertainment und *Everquest* [[httpEQ](http://EQ)] von Verant Interactive, die ersten drei großen Vertreter dieser neuen Marktnische. [FRIE02 S.8f]



**Abbildung 2.3** Screenshot aus dem MMORG EverQuest [[httpEQ](http://EQ)]



**Abbildung 2.4** Screenshot aus Dark Age of Camelot [[httpDAoC](http://DAoC)]

Letztgenannte Spiele zeichnen auch die Richtung vor, in die sich Persistent Worlds in den nächsten Jahren entwickeln werden. Mit *Dark Age of Camelot* [httpDAoC] von Mythic Entertainment und Funcom's *Anarchy Online* [httpAO] erscheinen um die Jahrtausendwende weitere solcher Spiele, die ebenfalls in einem Fantasyszenario platziert sind. Zu *Dark Age of Camelot* ist noch anzumerken, dass es jenes Onlinespiel ist, das den saubersten Launch aufgrund sehr weniger Bugs hinlegte [MULL02 S.339].

2003 erscheint mit *Star Wars Galaxies* zur Abwechslung auch ein Echtzeitstrategietitel in diesem von Rollenspielen dominierten Bereich. In weiterer Folge versucht beinahe jeder Spielehersteller auf den Zug der Persistent Worlds aufzuspringen. Dabei gelingt es jedoch nicht jedem ehrgeizigen Projekt auf dem Markt zu bestehen. Zu viele versuchen sich und scheitern, darunter *Ultima X* und *Warhammer Online* [httpWO]. Auf der anderen Seite entsteht die nächste Generation der großen MMO(R)Gs, *Everquest II* [httpEQ2], *World of Warcraft* [httpWoW], *Ragnarok* [httpRag], *Guild Wars* [httpGW], *Dungeon's & Dragons Online* [httpDDO] und noch einige mehr.

Bezüglich der Browserspiele beginnt die Geschichte eigentlich 1995 mit der Fertigstellung des Strategiespiels *SOL* [httpSOL], das erstmals eine persistente Welt für mehrere Spieler bietet. *SOL* spielt im Weltraum, jeder Spieler kontrolliert einen Planeten, den er ausbauen und verteidigen muss. Bei diesem Spiel gibt es noch keine Resets, sodass dieses Universum seit über zehn Jahren persistent ist. Der wirkliche Startschuss für die Browserspiele folgt im Jahr 2000 mit *Planetarion* [httpPLAN], das es bereits auf 175.000 Spieler pro Runde bringt. Im deutschsprachigen Bereich beginnt der Hype erst mit *Gangwars* [httpGANG] und *Galaxy Wars* [httpGXY]. Innerhalb von 5 Jahren entwickelt sich aber eine schier unüberschaubare Anzahl an Browserspielen.

## **2.2 (Flash) Browserspiele**

Für diese Spiele wird nur ein Webbrowser benötigt, um in die Onlinewelt einzutauchen. Anstatt eine Clientsoftware zu installieren, reicht es aus, sich auf der Webseite des jeweiligen Spiels anzumelden. Die Spiele laufen dann direkt im Browser des Spielers ab. Dadurch sind diese Spiele plattformunabhängig und unterscheiden sich – je nach Qualität der Programmierung – auch in den unterschiedlichen Browsern nur marginal. Im Endeffekt ist bei Browserspielen aber ohnehin die Spieltiefe viel wichtiger als die grafische Präsentation. [LOBER03]

Die Spiele laufen zeitlich begrenzt bzw. unbegrenzt oder sind rundenbasiert. Ein rundenbasiertes Spiel läuft in einer durch eine Zeitvorgabe oder ein bestimmtes Spielziel begrenzte Runde ab, in der der Spieler seine Aktionen planen und durchführen kann. Die Auswirkungen sind oft erst in den nächsten Runden erkennbar. Auf diese Weise werden die Spieler von dem Druck befreit, das Spielgeschehen über einen unbegrenzten Zeitraum überwachen zu müssen, um keine Rückschläge, zum Beispiel durch Angriffe von anderen Spielern, zu erleiden.

Da Browserspiele keine Software benötigen die verkauft werden kann, muss die Finanzierung eines solchen Spiels anderweitig erfolgen. Noch heute sind die meisten Projekte Non-Profit-Projekte, die bestenfalls die laufenden Kosten über Werbung, Googles AdSense-Programm oder den Verkauf von Premium-Accounts decken.

Da es meist das Credo der Entwickler ist, das Spiel der Internetgemeinschaft gratis zur Verfügung zu stellen, haben sich Premium-Accounts als kostendeckender Kompromiss herauskristallisiert. Damit ist gemeint, dass das Spiel zwar grundsätzlich kostenfrei gespielt werden kann, für mehr Komfort, keine Werbung und zuweilen auch spielerische Vorteile, ein entsprechender Account gekauft werden muss. Dieses System birgt jedoch die Gefahr, dass sich der Rest der Community, falls es zu einer unfairen Begünstigung zahlender Spieler kommt, schnell nach einer der zahlreichen Alternativen umsieht.

Der Trend geht jedoch auch dahin, dass sich immer mehr Projekte unter die Fittiche eines größeren Publishers begeben. Die kommerziellen Anbieter stellen Know-how sowie die finanziellen und personellen Ressourcen zur Verfügung, um den jeweiligen Titel besser am Markt zu positionieren. [LOBER06]

Im Vergleich zu dem clientbasierten Spielen wie *Dark Age of Camelot* oder dem MMORG-Blockbuster *World of WarCraft*, die Umsätze in Millionenhöhe generieren, befindet sich der Markt für Browserspiele jedoch noch in den Kinderschuhen. Allerdings verfügt dieser Markt über ein starkes Wachstum mit zurzeit hunderten verschiedenen Browserspielen, zu denen ständig mehr neue hinzukommen, als alte von der Bildfläche verschwinden. Schließlich haben Browserspiele meist auch eine weit längere Lebensdauer als ihre Offline-Pendants.

## **Interaktion**

Multiplayer-Onlinespiele leben hauptsächlich von der Interaktion der Spieler untereinander, wobei neben Kommunikation und dem gegenseitigen Bekämpfen auch Wirtschaftssysteme eine tragende Rolle spielen. Jedes dieser Spiele verfügt in einem gewissen Umfang über Güter, die man sammeln und einsetzen kann. Dies können Ressourcen bei Strategiespielen oder Ausrüstungsgegenstände bei Rollenspielen sein. Diese sind entweder langwierig oder schwierig zu sammeln bzw. herzustellen und bieten die Basis für einen Onlinehandel. Ein derartiges Wirtschaftssystem unterliegt einerseits, was Angebot und Nachfrage betrifft, den gleichen Regeln wie die reale Wirtschaft. Auf der anderen Seite gibt es spielspezifische Besonderheiten, die dem Spielspass dienen. Beispielsweise wäre es zwar realistisch, wenn die Preise für Nahrungsmittel bei Ultima Online je nach Nachfrage steigen oder fallen würden, auf der anderen Seite würde dies jedoch dazu führen, dass Spieler zuviel Zeit mit dem Geldverdienen für solche Grundbedarfsmittel verbringen würden und sich weniger auf das eigentliche Spiel konzentrieren können. In einigen Spielen wurde versucht geschlossene Wirtschaftskreisläufe zu implementieren, was sich auf Dauer jedoch nicht durchgesetzt hat, da ein solches System äußerst komplex und schwer zu entwickeln ist. Außerdem besteht immer die Gefahr, dass wichtige Güter knapp

werden, dadurch die Preise steigen und vor allem neue Spieler keine Chance mehr haben, im Spiel weiterzukommen. Es muss jedoch immer darauf geachtet werden, dass es einen nie versiegenden Strom neuer, attraktiver Güter gibt, für die erfahrene, also reiche, Spieler ihr Geld verwenden können. Vor allem Geld unterliegt in Onlinespielen einer gewissen Inflation, da jene, die viel Zeit in solchen Welten verbringen, Unsummen davon ansammeln können. Deshalb gehen die Spieler oft zum Tauschhandel über, wobei sich Schattenwährungen herausbilden. Beispielsweise wurden bei *Diablo II* eine Zeitlang die Preise für begehrte Gegenstände in SoJ angegeben (Stone of Jordan, ein seltener Ring). [LISCH03]

## Genres

Die von Browserspielen verwendeten Genres sind nicht besonders breit gefächert. Ein Großteil dieser Spiele sind (Echtzeit-)Strategiespiele und Sportmanager, garniert mit einer Handvoll Rollenspiele. Letzteres Genre genießt bei den kommerziellen Multiplayer-Spielen eine große Popularität, wird im Bereich der Browserspiele jedoch eher stiefmütterlich behandelt. Grund dafür ist mit hoher Wahrscheinlichkeit der große technische, wie grafische Aufwand und die Notwendigkeit eine vielfältige, glaubhafte Spielwelt zu schaffen, wohingegen man Strategiespieler auch mit zufällig auf einer Karte verteilten Planeten und einer Handvoll Einheiten befriedigen kann.

An dieser Stelle sollen auch diese drei Hauptgenres kurz erörtert werden. Strategiespiele unterscheidet man nach dem Spielzweck in Kampf- bzw. Kriegsspiele und Aufbautitel. Die meisten strategieorientierten Browserspiele bieten beides, indem man seine Burg, seine Basis oder seinen Planeten zuerst mit verschiedenen Gebäuden aufrüstet und mehr oder weniger komplexe Warenkreisläufe in Gang bringt, um genügend Militär zum Angriff bzw. zur Verteidigung aufzubauen zu können.

Sportmanagerspiele haben meist auch einen gewissen Aufbau- bzw. Wirtschaftspart. Beispielsweise kann man sein Stadion ausbauen um mehr Geld zu verdienen, mit dem man wiederum bessere Spieler kaufen kann.

Hauptaufgabe ist jedoch das Managen eines Sportvereins. Es müssen Spieler gehandelt, trainiert und richtig aufgestellt werden, Motoren getuned und Fahrer gekauft, oder Pferde ersteigert und Jockeys ausgewählt werden – je nach Sportart eben.

Das Rollenspielgenre ist bei den Browserspielen weniger in der *Diablo*-Hack'n'Slay-Manier vertreten, sondern meist dadurch, dass man einen in der Realität eher ungewöhnlichen Beruf ergreift und versucht möglichst viel Geld und Ruhm einzustreichen, indem man verschiedene Aktionen ausführt.

## Marktübersicht

Das Onlinespiele-Portal *gamingfacts* [httpGF] listet mit Stichtag 26. Juni 2006, 345 Browserspiele, wobei Science-Fiction-Szenarien mit 118 Titeln (34%) die größte Gruppe darstellen. Ziemlich gleichauf folgen Mittelalter- und Fantasyszenarien (78 Spiele / 23%) sowie die Kategorie „Gesellschaft“ (92 / 27%). *gamingfacts* fasst in dieser Kategorie Spiele zusammen, in denen man eine Rolle übernimmt, beispielsweise die eines Nachtclubbesitzers, eines Immobilienhais oder eines Steinzeitmenschen. Ziel dieser Spiele ist es, jeweils der Beste in seinem Metier zu werden.

47 Spiele (14%) entfallen auf den Sportbereich, der zu knapp 90% aus Managerspielen besteht. Die abgedeckten Sportarten werden klar von Fußball dominiert (53%), gefolgt von Motorsport (17%) und mit jeweils weniger als vier Titeln: Eishockey, Reiten, Handball, Tennis und Hockey. Als Nicht-Managerspiele gibt es nur einfache Autorennen auf Flashbasis sowie ein einzelnes Schachspiel. Zu den Spielerzahlen der einzelnen Spiele existieren kaum gesicherte Informationen. Oft wird einfach die Anzahl aller erstellten Accounts angegeben, womit auch alle mitgezählt werden, die nur zum Antesten des Spiels angelegt wurden und nicht mehr aktiv sind.

Die Aussicht, alleine oder mit einem sehr kleinen Team und verhältnismäßig geringem Aufwand ein populäres Spiel zu entwickeln, sorgt dafür, dass es immer wieder Nachschub gibt. Die Szenarien sind freilich größtenteils dieselben: Strategie in verschiedensten zeitlichen Umgebungen, hauptsächlich Mittelalter und

Science-Fiction im Weltraum. Mittlerweile entwickeln sich jedoch vermehrt alternative Szenarien, z.B. World of Crime [[httpWoC](http://WoC)], in dem man mit einer Menge Geld startet und dieses – durch reichlich illegale Geschäfte – noch vermehren muss. Die Statistik zeigt auch die Übermacht der Fußballmanager im Bereich Sport. Dieser Boom ist möglicherweise auch auf die Weltmeisterschaft in Deutschland zurückzuführen, vor allem, da es einige Spiele gibt, die dieses Ereignis zum Inhalt haben.

Verschiedene Interviews mit den Entwicklern von Browserspielen auf *gamingfacts* zeigen auch auf, wie sich der Markt entwickelt. Die meisten derer, die selbst ein Browserspiel entwickeln, haben davor einige Zeit mit bestehenden Spielen verbracht. Viele programmieren dann einfach ein vorhandenes Spiel nach, ändern Features oder das Setting und fügen neue Spielelemente hinzu. Dadurch ergibt sich eine große Masse ähnlicher Spiele.

Auf der anderen Seite bietet diese Art von Computerspielen eine relativ gute Basis für innovative Spielkonzepte. Während im klassischen Spielmarkt Spiele mit enormen Aufwand und Entwicklungskosten produziert werden, ist das Risiko bei einem Browserspiel weitaus geringer. Hier gibt es noch die Möglichkeit, quasi in der Garage, ein Spiel zu entwickeln, das genau dem entspricht, was man von anderen Spielen schon immer erwartet hat.

Das Spieleportal *gamingfacts* führt auch Interviews mit den Entwicklern verschiedener Browserspiele. Bezogen auf Teamgröße, Entwicklungsaufwand und Finanzierung lassen sich daraus einige Informationen ableiten.

Wie bereits erwähnt, werden die meisten Browserspiele von einem einzelnen Programmierer entwickelt, der oft noch von einem Grafiker unterstützt wird. Im späteren Verlauf kommen dann Hilfskräfte in Form von Foren-Administratoren, Testern und so genannten Multi-Huntern hinzu. Letztere kümmern sich um das Aufspüren von Spielern, die mehrere Accounts angelegt haben um sich Spielvorteile zu erschleichen. Die wenigsten Entwicklungen verfügen von Beginn an über vollständige Teams aus Entwicklern, Grafikern und Testern.

Die Angaben zu den Entwicklungszeiten sind unvollständig und auch wenig aussagekräftig. Da die Spiele meist neben Studium oder regulärer Arbeit

entwickelt werden, bleibt unklar wie viel Zeit effektiv in das Spielprojekt fließt. Grundsätzlich zeichnet sich ab, dass sich Browserspiele innerhalb von zwei bis vier Monaten zu einer Beta-Version entwickeln lassen. Diese öffentlichen Tests nehmen meist noch einige Wochen bis Monate in Anspruch, bevor dann der offizielle Spielbetrieb aufgenommen werden kann.

An dieser Stelle muss auch festgehalten werden, dass die Entwicklung eines solchen Spiels mit dem Onlinestellen noch lange nicht abgeschlossen ist, sondern vielmehr eigentlich erst beginnt. Es ist enorm wichtig das Spiel ständig weiterzuentwickeln, wobei auf das Feedback der Community geachtet werden sollte. Oft ist es auch nötig, dem Spiel zusätzliche Inhalte hinzuzufügen oder die Grafik bzw. das Benutzerinterface zu verbessern. So gaben auch einige Entwickler an, bereits mehrere Jahre – zumindest teilweise – mit dem Projekt beschäftigt zu sein.

Die Bereitschaft so viel Zeit in ein Projekt zu investieren begründen die meisten mit Passion für Spiele und deren Programmierung bzw. betrachten es als Hobby. Dazu passt auch, dass alle der Befragten angaben, auf freiwilliger, ehrenamtlicher Basis zu arbeiten. Manche finanzieren auch die laufenden Kosten (Server, Traffic, etc.) aus eigener Tasche, während die meisten – zumindest zusätzlich – auf Einnahmen durch Werbebanner und den Verkauf von Premium-Accounts setzen. Ein kleiner Teil der Browserspiele verwendet auch Googles AdSense als Einnahmequelle. Allen gemein ist die Tatsache, dass die Spiele grundsätzlich kostenlos zu spielen sind. Premium-Accounts bieten meist nur Komfort-Verbesserungen, sind also nicht notwendig um in der Spielwelt bestehen zu können.

## **2.3 Aufbau von Onlinespielen**

Dieses Kapitel beinhaltet Beschreibungen des Aufbaus von und der angewandten Techniken in Onlinespielen sowie Erläuterungen zu den verschiedenen Möglichkeiten den Spielablauf eines solchen Spiels zu strukturieren. Bei den Techniken wird auf die Kommunikation zwischen den Spielern bzw. mit dem Server eingegangen, außerdem auf Fehlerbehandlung und Vorhersage-Algorithmen um mit den meist zu hohen Latenzzeiten umzugehen.

Bei der Beschreibung von MMOGs ist ein Begriff klar herauszustreichen: Persistent World (PW). Diese Art der Spielwelt existiert nicht wie in klassischen Spielen für jeden Spieler einmal, sondern einmal für tausende Spieler. Die Welt wird auf einem Server bereitgestellt und jeder der sich in das Spiel einwählt bekommt einen Blick auf diese Welt präsentiert. Alle Aktionen die man setzt haben unmittelbaren Einfluss auf die gesamte Welt und damit – bis zu einem gewissen Grad – auch auf alle anderen Spieler. Damit ist dieses Prinzip sehr nahe an der realen Welt, in der ebenfalls Dinge geschehen, während man schläft. Ein wichtiger Aspekt ist auch das Fehlen jeglicher Backup-Option in solchen Spielen. Man kann zwar seinen Status speichern und an einem anderen Tag wieder weiterspielen, es ist jedoch nicht möglich verschiedene Spielstände anzulegen und eine erfolglose Aktion ungeschehen machen, indem man auf einen früheren Speicherpunkt zurückgreift.

Bezüglich des Spielablaufs tritt bei Multiplayer-Spielen mit der Interaktion zwischen den Spielern ein Problem auf, das den klassischen Singleplayer-Spielen fremd ist: die Behandlung gleichzeitiger Aktionen mehrerer Teilnehmer. Während in einem klassischen Spiel dem Spieler praktisch unbegrenzt Zeit für das Setzen seiner Aktionen zugestanden wird – natürlich limitiert durch das Verhalten von Gegnern, verschiedener Spielereignisse und der Gewichtung des Spiels auf die Reaktionsschnelligkeit des Spielers im Allgemeinen – sind bei Multiplayer-Spielen, vor allem bei solchen im Onlinebereich, Einteilungen im Spielablauf nötig.

## turn based

In rundenbasierten Spielen wird der Spielfluss in abgegrenzte Schritte, so genannte Runden, eingeteilt. Eine Runde kann dabei sowohl eine definierte Zeitspanne sein, die man maximal für seinen Zug benötigen darf, als auch einfach dann enden, wenn alle teilnehmenden Spieler ihre Aktionen durchgeführt haben.

Hierbei werden zwei Typen unterschieden, je nachdem ob die Spieler innerhalb einer Runde gleichzeitig (phase-based) oder nacheinander ihre Züge spielen können. Die zweite Art wird außerdem in folgende Untergruppen unterteilt:

- Ranked                    der startende Spieler ist jede Runde derselbe
- Round-robin            der startende Spieler wird jede Runde reihum gewechselt
- Random                    der startende Spieler wird jede Runde neu ausgelost
- Initiative-based        der startende Spieler wird anhand eines Wertes bestimmt

Bei phase-based-Spielen wird außerdem noch unterschieden ob pro Zug eine (clock-based) oder mehrere Aktionen gesetzt werden können. Bei zweitem Typ wird ein Aktionspool verwendet, den man mehrere Runden lang füllen kann, indem man keine oder weniger Aktionen durchführt als möglich, um dann viele Aktionen innerhalb einer Runde zu setzen. [httpWIKI]

Vor allem im Offline-Bereich gibt es auch Spiele, die, wie beispielsweise *Fallout* oder *Baldur's Gate*, nur in manchen Bereichen – meist der Kampfphase – rundenbasiert, ansonsten jedoch in Echtzeit ablaufen.

Zu rundenbasierten Spielen zählen auch E-Mail-Spiele, die, wie z.B. Schach im Europa des 17. Jahrhunderts, per Brief, respektive über E-Mail, gespielt werden.

## **tick based**

Diese Art von Spiel wird in Zeiteinheiten (ticks) unterteilt, die, je nach Spiel, von Sekunden über Tage bis hin zu Monaten dauern können. Sie werden hauptsächlich in Browserspielen verwendet und definieren Zeitpunkte, an denen verschiedene Aktionen stattfinden. Im Gegensatz zu rundenbasierten Spielen finden Ticks immer nach gleichen Zeitabschnitten statt. So wird bei Spielen mit einem tick von 30 Minuten, jede halbe Stunde der Kontostand des Spielers aktualisiert, berechnet wie viel Energie die Kraftwerke der einzelnen Spieler produziert haben oder der Charakterlevel in Rollenspielen erhöht.

Je geringer die tick-Zeit desto schneller läuft das Spiel im Allgemeinen ab. Ticks unter 10 Minuten werden als Speed-Runden bezeichnet, die mehr Spielspaß bringen können, da man schneller bzw. öfter Aktionen setzen kann, auf der anderen Seite jedoch weit höhere Anforderungen an den Server hinsichtlich Performance und Bandbreite stellen.

Nach dem Überblick über die unterschiedlichen Arten der Strukturierung des Spielablaufs soll im folgenden Abschnitt auf die diversen Spieler-Spieler- bzw. Client-Server-Kommunikationsarten eingegangen werden.

## **Kommunikation**

Die Kommunikation zwischen den einzelnen Spielern kann je nach Spiel sehr unterschiedlich implementiert sein. Man unterscheidet zwischen einer asynchronen Kommunikation, bei der ein Spieler Informationen über seine Aktion an den Server sendet, der diese wiederum speichert und an alle anderen Clients erst ausliefert, wenn diese ein Update der Spielwelt erfragen, und der Verwendung eines Kommunikationsservers, der eventbasiert, Informationen an alle Spieler ausliefert sobald sich in der Spielwelt etwas getan hat. Browserspiele verwenden hier meist eine asynchrone Kommunikation, da die Verwendung eines Kommunikationsservers mit erheblich mehr Aufwand und zum Teil auch Kosten verbunden ist.

Die Kommunikation zwischen dem Client bzw. Webbrowser und dem Server unterscheidet sich ebenfalls zwischen Browserspielen und kommerziellen, clientbasierten MMOGs.

Während clientbasierte Spiele Informationen über den Spielstatus der einzelnen Teilnehmer über das schnelle UDP (user datagramm protocol) übertragen und durch die fehlenden Bestätigungen dieses Protokolls erheblich geringere Paketgrößen und Antwortzeiten erreichen, verwenden die Browserspiele das Internetprotokoll TCP/IP mit damit einhergehender geringerer Übertragungsgeschwindigkeit. Darauf aufbauend werden entweder XML-Pakete oder einfach nur Variablen über GET oder POST übermittelt. Manche Entwickler verwenden vor allem bei Echtzeitspielen, die besonders auf schnelle Übertragung angewiesen sind, proprietäre Schnittstellen um die Paketgrößen möglichst gering zu halten.

## **2.4 Communities / Virtual Societies**

Wie in der realen Welt bilden sich auch in einer virtuellen Umgebung Gemeinschaften heraus, die auf gemeinsamen Zielen basieren. Es ist nun nicht so, dass alle Spieler in einer PW einer einzigen großen Community angehören. Dieser Fall wäre auch weitestgehend sinnlos, da eine solche Gemeinschaft nicht mehr organisierbar wäre. Vielmehr entwickeln sich in der Spielerbasis *micro-communities*, wie beispielsweise Gilden, die von einigen wenigen Spielern angetrieben werden. Diese präsentieren sich auch auf eigenen Webseiten mitsamt Foren, IRC und Newsbereichen. Solche Gemeinschaften bilden sich oft auch aufgrund von spielinternen Gemeinsamkeiten, wie Charakterklassen, Servertypen (z.B. PvP), vom Spiel vorgegebenen Fraktionen oder Erfahrungsstufen. Viele Spiele unterstützen solche Communities meist, indem Möglichkeiten für Auseinandersetzungen großer Gruppen geschaffen werden. Diese *micro-communities* werden wiederum von einzelnen oder kleinen Gruppen von Spielern angeführt.

Communities sind für Onlinespiele von außerordentlicher Bedeutung. Kommunikation zwischen Spielern und das Eingebundensein in eine Gemeinschaft tragen einen großen Teil zum Spielerlebnis bei. Somit können auch Spieler die möglicherweise vom Spielinhalt nicht so überzeugt sind für ein Spiel gewonnen werden, nur aufgrund einer aktiven Community und dem Umgang der Spieler untereinander. Schlussendlich entscheidet auch die Stärke der Community über die Überlebensfähigkeit des jeweiligen Spiels auf dem Markt. Spieler die sich in einer Gemeinschaft wohl fühlen, werden viel weniger von anderen Spielen angezogen werden, als solche, denen der soziale Part bei einem Spiel fehlt. [FRIE02]

Online-Communities sind wie Gesellschaften der realen Welt von großer Dynamik geprägt und unterliegen ebenso Darwin's Evolutionstheorie. Sie verändern sich ständig und passen sich an um möglichst lange zu überleben. Die Community eines Spiels entwickelt ihre eigenen Regeln, Werte und Normen und bietet ihren Mitgliedern auch ein Forum dessen Themen über die des Spiels

hinausgehen. Durch diesen breit gefächerten Gedankenaustausch bilden sich starke Bindungen innerhalb der Community. [FRIE02]

Auch die Geschichte eines Spiels wird von der Community geschrieben. Im Gegensatz zu klassischen Offlinespielen, in denen der Spielverlauf von den Entwicklern vorgezeichnet und vom Spieler angenommen und verfolgt wird, bieten Onlinespiele nur ein Framework und gewisse Spielmechaniken innerhalb deren Grenzen die Spieler die Geschichte der Welt fortschreiben. Hier müssen den Spielern, vor allem in Rollenspielen, auch möglichst viele Möglichkeiten gegeben werden sich in die Spielwelt einzubringen und gemeinschaftliche Events zu erleben.

Die Community eines Onlinespiels lebt auch nicht nur in der Spielwelt sondern vielmehr auch außerhalb davon. Nur ein geringer Teil der Aktivitäten einer solchen Gemeinschaft findet innerhalb der Grenzen der Onlinewelt statt, der viel größere Teil der Kommunikation der Mitglieder auf Webseiten, über E-Mail, Chat und Telefon. Diese Eindrücke wiederum wirken sich auf das Erlebnis des Spielers in der tatsächlichen Spielwelt aus und entscheiden ob er bei einem Spiel bleibt oder nicht. Schließlich kann das Bedürfnis von Spielern soweit gehen, die anderen Spieler mit denen sie monatelang eine Online-Beziehung geführt haben auch in der realen Welt zu treffen. Dies wiederum kann von den Spielentwicklern genutzt werden um die Community zu stärken, indem Treffen oder Events organisiert werden, die den Spielern diese Möglichkeit bieten.

## **2.5 Spielertypen**

Bei der Entwicklung eines Onlinespiels, im besonderen einer PW, ist es unerlässlich einen Blick auf die verschiedenen Typen von Spielern zu werfen, um entscheiden zu können in welche Richtung man seine Onlinewelt ausrichten möchte und um herauszufinden, welche Aspekte notwendig sind um eine stabile virtuelle Gesellschaft zu schaffen. Obwohl Richard A. Bartle's Ausführungen, die die Grundlage für dieses Kapitel bilden, für die MUDs Mitte der 90er Jahre des vorigen Jahrhunderts gedacht waren, haben sie heute genauso Gültigkeit und trotz der Fokussierung auf clientbasierte, „professionelle“ Onlinespiele, haben sie ebenso Bedeutung für die Entwicklung von Browserspielen.

Zuerst gilt es herauszufinden welche Ziele sich innerhalb der Spielumgebung ergeben. Es handelt sich dabei um vier dedizierte Ziele, wobei jeder Spieler eines besonders und die anderen vermindert, meist um das Primärziel zu erreichen, verfolgt.

### **Errungenschaften innerhalb der Spielumgebung**

Dabei setzt sich der Spieler spielbezogene Ziele, wie z.B. das Erreichen eines bestimmten Status, das Sammeln von besonderen (Ausrüstungs-) Gegenständen oder Erfahrung. Dieses Ziel ist klar auf das Spiel abgestimmt und um diese Spieler zu erreichen muss die Onlinewelt Hierarchien zur Verfügung stellen und die Sammelwut der Spieler befriedigen.

### **Erkundung des Spiels**

Hier geht es nur zu Beginn um die Erforschung der Spielumgebung, viel wichtiger ist bei diesem Ziel das Verständnis für die Spielmechanik und Reaktionsketten innerhalb des Spiels. Schließlich geht es dabei um das Ausprobieren verschiedenster Aktionen, mitunter auch um Bugs, Abkürzungen oder Besonderheiten herauszufinden, aber vor allem um ein umfassendes Wissen über das betreffende Spiel anzusammeln.

### **Interaktion mit anderen Spielern**

Dieses soziale Ziel will durch Nutzung der Kommunikationsmittel, die vom Spiel angeboten werden, erreicht werden. Dieses Ziel geht auch weit über die standardmäßige Kommunikation zum Zweck des Zusammenspielens hinaus. Vielmehr geht es hierbei um Kommunikation im herkömmlichen Sinn, um Gespräche über alltägliche Themen und die Vorgänge in der realen Welt.

### **Gegner besiegen**

Das einzige Ziel bei dieser Spielart ist das Besiegen anderer Spieler. Dabei werden alle Möglichkeiten, die das Spiel zu diesem Zweck bereitstellt – wenn das Töten anderer Spieler vom Spielprinzip überhaupt vorgesehen ist – genutzt um dieses Ziel zu erreichen. In seltensten Fällen werden diese Mittel auch eingesetzt um schwächeren Spielern zu helfen.

Aus diesen Zielen ergeben sich nach Bartle folgende vier Spielertypen:

### **Achiever**

Dieser Typus sucht hauptsächlich nach Schätzen, sammelt Gegenstände und versucht Erfahrungspunkte zu sammeln um einen höheren Level zu erreichen. In gewisser Weise betätigt er sich natürlich auch als Entdecker, um die Schätze und Gegenstände zu finden und um die Spielmechanik genug zu verstehen um seine Ziele zu erreichen. Auch kommuniziert er mit anderen Spielern um von deren Wissen über das Spielsystem zu profitieren und kämpft gegen andere Spieler um Rivalen zu beseitigen oder, wenn dies vom Spiel honoriert wird, um viele (Erfahrungs-)Punkte zu bekommen.

## **Explorer**

Diese versuchen die Spielwelt in allen Einzelheiten zu erkunden und zu verstehen. Der Entdecker versucht auch alle möglichen Aktionen an ungewöhnlichen Orten um Bugs zu finden. Er sammelt Punkte nur um andere Bereiche des Spiels frei zu schalten und kommuniziert mit anderen Spielern um Informationen zu erhalten. Er greift andere Spieler nicht an, bekämpft sie aber zur Selbstverteidigung. Sein vornehmliches Ziel ist das Sammeln von Wissen über das Spiel.

## **Socializer**

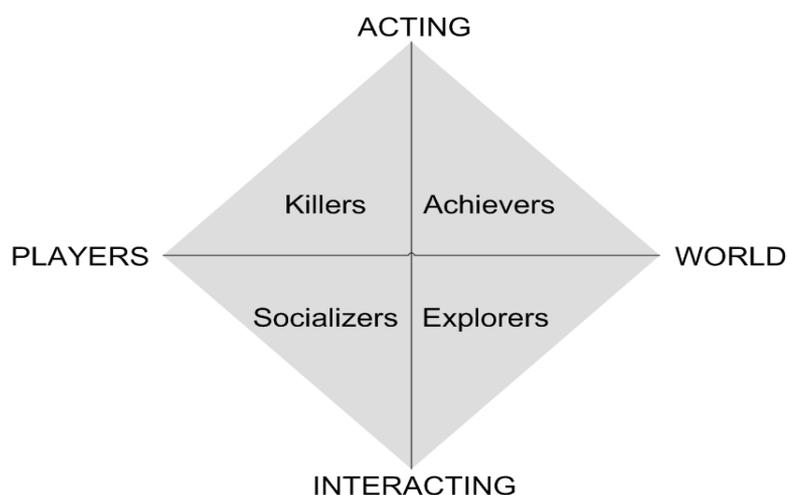
Diese wiederum suchen den Kontakt zu anderen Spielern, bilden Allianzen und versuchen gemeinsam etwas zu erreichen. Zwangsläufig müssen sie auch das Spiel erkunden um an den Gesprächen über die Spielwelt partizipieren zu können und müssen auch Punkte sammeln um alle Kommunikationsmittel, die möglicherweise nur für erfahrene Charaktere zur Verfügung stehen, nutzen zu können und um einen Status in der Community zu erreichen. Hauptsächlich wollen sie andere Spieler kennen lernen und Beziehungen aufbauen. Für diese Art Spieler sind virtuelle Welten auch ein Ersatz für reale soziale Kontakte.

## **Killer**

Das Hauptziel dieses Typus ist das Besiegen anderer Spieler, weswegen sich mittlerweile auch die Bezeichnung *player killer* (pk) durchgesetzt hat. Er versucht dieses Ziel durch Überlegenheit der eigenen Spielfigur, aber auch durch Tricks und das Ausnutzen von Bugs im Spiel zu erreichen. Auch diese Spieler müssen Punkte sammeln um stark genug zu werden um andere Spieler zu töten. Auch entdeckt er Spielmechaniken um immer neue Möglichkeiten zu haben sein Ziel zu erreichen.

Für den dauerhaften Bestand einer Onlinewelt ist es wichtig diese Spielertypen auszubalancieren. Zu viele oder zu wenige einer Sorte sorgen dafür, dass auch andere das Spiel verlassen. Beispielsweise sorgen zu viele „Killer“ dafür, dass die Sammler ausgerottet werden. Umgekehrt führt ein Mangel an Entdeckern dazu, dass die Sammler nicht wissen wonach sie suchen sollen. Werden generell weniger Spieler, verlassen auch die „Socializer“ das Spiel, womit die Hauptbeute der Jäger wegfällt und auch diese in eine andere virtuelle Welt abwandern.

Die folgende Abbildung zeigt die Interessensphären der einzelnen Spielertypen. Die x-Achse geht von einer Fokussierung auf andere Spieler zu einem vornehmlichen Interesse an der Spielumgebung. Die y-Achse reicht vom Spielen mit anderen zum Spielen gegen bzw. ohne andere.



**Abbildung 1.5** Verteilung von Spielertypen bezüglich Interessensphären (vgl. [BARTLE 1996])

Wie genau die Balance zwischen den einzelnen Gruppen aussieht, hängt vom jeweiligen Spiel ab, gewollte Änderungen in alle Richtungen sind möglich, führen aber nicht zwangsläufig zum gewünschten Ergebnis.

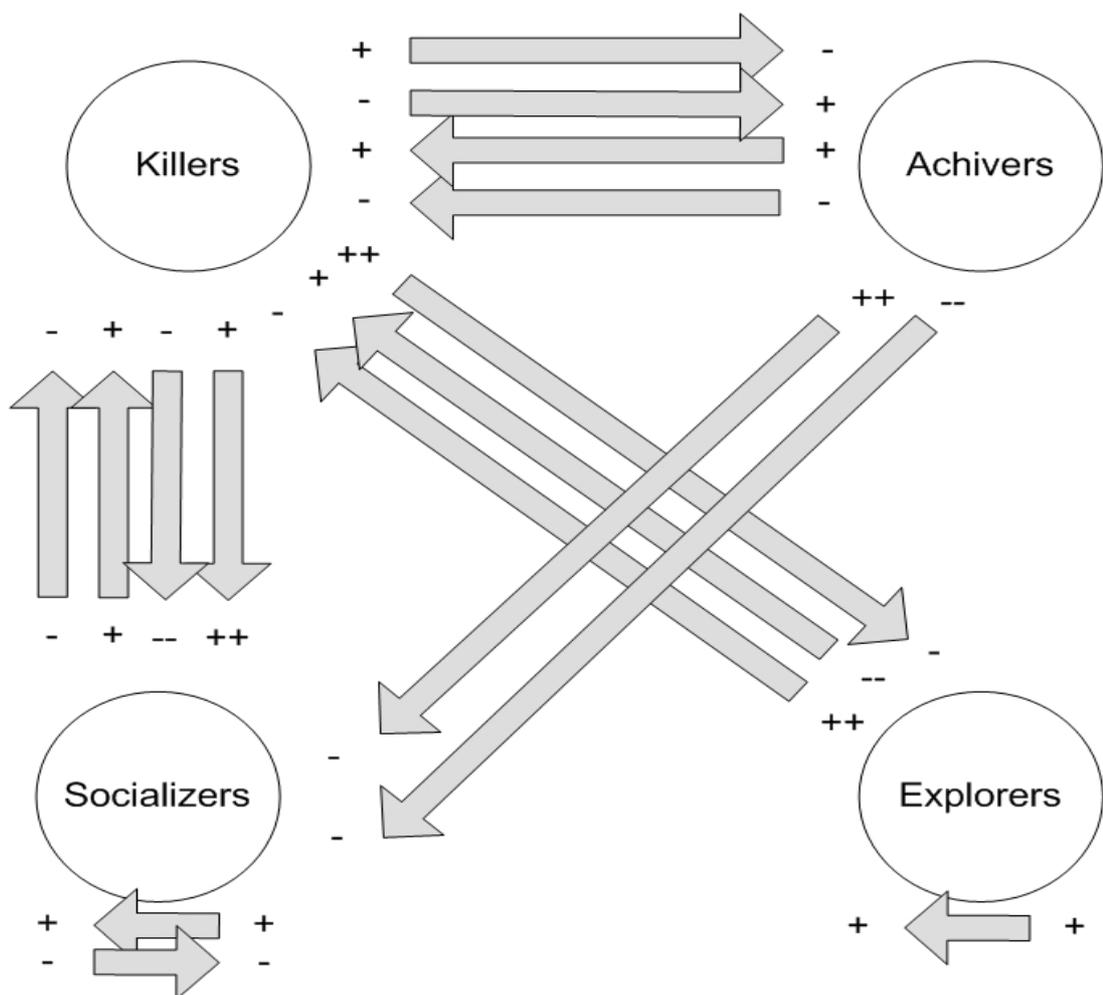
Ein höherer Fokus auf die Spieler kann erreicht werden, indem man im Spiel viele Möglichkeiten der Kommunikation bereitstellt aber kaum etwas anderes. Je weniger Spiel daneben vorhanden ist, desto mehr entsteht ein Chat und das Ganze hört auf ein Spiel zu sein.

Zu erzwingen, dass sich alles um die Spielwelt dreht, ist ebenso einfach. Es reicht, die Welt besonders groß auszulegen, die Spieler weit voneinander entfernt

zu platzieren und die Interaktionsmöglichkeiten, sollten sich Spieler trotz allem treffen, zu minimieren. Im Endeffekt geht aber dadurch mit der sozialen Komponente ein wichtiger Aspekt von MMOGs verloren und man kann genauso ein Singleplayer-Spiel spielen.

Geht man stark in Richtung Interaktion zwischen Spielern und verzichtet darauf, den Spieler spielen zu lassen, führt es dazu, dass sich das Spiel wie ein Film anfühlt. Andererseits führt ein Verringern der Interaktion zwischen Spielern dazu, dass jeder immer wieder nur die gleichen Aufgaben ausführt um Punkte zu bekommen und das Spiel wird schnell langweilig.

Abbildung 2.6 zeigt, wie Änderungen an der Größe der Gruppe sich auf die Population einer anderen Gruppe auswirken.



**Abbildung 2.6** Abhängigkeiten der Spielertypen untereinander (vgl. [BARTLE 1996])

Die Plus- und Minuszeichen an den Pfeilenden zeigen einen Zuwachs bzw. eine Schrumpfung, doppelte Zeichen eine entsprechend starke Veränderung. Erhöht sich beispielsweise die Anzahl an „Killern“, schrumpft die Anzahl der Sammler [BART96].

Im Endeffekt dreht sich alles um die richtige Balance. Diese ist auch die größte Herausforderung beim Entwickeln und Erhalten einer Onlinewelt. Natürlich ändern sich die Anforderungen abhängig vom betrachteten Genre, die Kernaussage bleibt jedoch gleich. Ebenso wie in Rollenspielen, gibt es diese vier Gruppen von Spielern auch in dem, in Browserspielen weit verbreiteten, Genre der Strategiespiele. Auch hier gibt es Achievers, die versuchen eine möglichst starke Position aufzubauen und die Spielziele zu erreichen, genauso gibt es Killer, deren einziges Ziel die Vernichtung benachbarter Gegner ist. Socializer nutzen das Spiel als Forum bzw. Ersatz für das richtige Leben und schmieden Allianzen um mit anderen Spielern ins Gespräch zu kommen. Nicht zuletzt gibt es auch in einem Browserspiel viele Feinheiten der Spielmechanik zu entschlüsseln, sodass für Explorer genügend Beschäftigung vorhanden ist.

Beachtet man diese Überlegungen und stellt für jeden der einzelnen Typen Möglichkeiten zur Selbstverwirklichung zur Verfügung, ist eine gute Basis für einen längerfristigen Erfolg des Projekts geschaffen.

## 2.6 Designprinzipien

Grundsätzlich gibt es keinen Bauplan für ein erfolgreiches Onlinespiel, aber es gibt durchaus Eckpunkte an die man sich halten kann um seine Chancen zu verbessern. [KOST97] führt hier einige Regeln auf:

- es muss mehrere Wege geben im Spiel etwas zu erreichen
- es sollte möglichst einfach sein zwischen diesen Pfaden zu wechseln (möglichst ohne neu beginnen zu müssen)
- die Wege müssen in Meilensteine unterteilt sein (lieber weniger dafür signifikante)
- es muss so aussehen, als ob es immer neue Meilensteine gibt, als ob es immer etwas zu erreichen gibt

Neben diesen Prinzipien bezüglich des Weiterkommens im Spiel und der Motivation weiterzuspielen, sollte es auch so viele Arten wie möglich geben, sich auszudrücken, sein virtuelles Alter Ego zu gestalten. Das können beispielsweise unterschiedliche Charakterklassen wie Krieger oder Zauberer sein, oder auch die Unterstützung unterschiedlicher Spielweisen.

Wichtig ist auch dem PW-Ansatz treu zu bleiben und die Onlinewelt ständig verfügbar zu halten, um nicht das Vertrauen der Spieler zu verlieren. Überhaupt ist es von eminenter Bedeutung sich auf aktive Spieler zu konzentrieren, anstatt nur zu versuchen neue Spieler zu gewinnen. [JUD01]

Ein wichtiger Aspekt ist auch die Verwendung von statischen Elementen. Dabei ist es unerheblich, ob es sich dabei um eine vordefinierte Hintergrundgeschichte oder ein Rätsel, das immer gleich zu lösen ist, handelt. Man muss sich darüber im Klaren sein, dass sämtliche statischen Elemente binnen weniger Tage nach Veröffentlichung des Spiels bekannt, bzw. gelöst sein werden und über die üblichen Internetquellen verbreitet werden. Das Wissen über statische Elemente darf somit nicht das Spielerlebnis beeinflussen. Ideal wäre es eine Art Simulation zu schaffen, in der keine vordefinierten Geschichten erzählt werden, sondern den Spielern vielmehr die Freiheit gegeben wird, selbst ihre Geschichte zu schreiben.

Da dieser Ansatz jedoch um ein Vielfaches schwerer umzusetzen ist, gibt es auch keine Onlinewelt, in der das funktioniert. [KOST97]

Beim Erstellen einer Onlinewelt muss man sich auch bewusst sein, dass die Spieler überzogene Erwartungen an ein solches Spiel stellen: sie wollen für alles belohnt werden, wollen geschützt werden und wollen, dass ihre Probleme sofort gelöst werden. Außerdem werden sie ständig neue Features haben wollen, egal wie viele man ihnen gibt. Genauso wie mit den Erwartungen, ist es auch mit den Möglichkeiten, die man im Spiel anbietet. Auch hier wird man einen Kompromiss finden müssen zwischen großer Variation auf der einen Seite und der Vermeidung inkompatibler Spielelemente auf der anderen Seite.

Auch die „Wirtschaft“ in einem Onlinespiel ist nicht einfach am Leben zu erhalten. Es muss ständig neue Gegenstände oder sonstige Objekte geben, die man erreichen kann, da das Spiel sonst schnell langweilig wird, weil ohnehin jeder mit derselben Ausrüstung bestückt ist.

Eine gute Möglichkeit Spieler in der eigenen Welt zu halten, ist auch, ihnen eine Form von Besitz oder Verantwortung innerhalb der Welt zu geben. Soziale Banden verlagern sich meist hinaus aus dem Spiel, mit dem Effekt, dass am Ende nicht ein Spieler, sondern eine ganze Gruppe in eine andere Onlinewelt umzieht. Gibt man den Spielern jedoch etwas, das sie an die Spielwelt bindet, beispielsweise ein selbst gebautes Gebäude oder einen Posten mit Unterebenen, wird sie das länger im Spiel halten. [KOST97]

Zusammenfassend zeichnet sich folgendes Bild eines idealen Onlinespiels ab:

- es können unterschiedlichste Ziele auf unterschiedlichste Weise erreicht werden
- es gibt viele Möglichkeiten mit anderen Spielern zu interagieren
- die Geschichte des Spiels bildet nur den Rahmen, innerhalb dessen die Spieler ihre eigene Geschichte entwickeln
- es gibt für jeden Spielertyp reichlich Möglichkeiten sich zu verwirklichen
- die Onlinewelt ist umfang- und abwechslungsreich, wird ständig erweitert und bindet den Spieler ein (über Besitztümer, Ämter, Möglichkeiten etwas nachhaltig zu verändern)
- es ist für das Spielerlebnis unerheblich ob man zwei oder 20 Stunden pro Woche in der Spielwelt verbringt

### 3 Proof of Concept

Dieses Kapitel behandelt den praktischen Teil der Arbeit und beschreibt die einzelnen Projektphasen von der Idee zu TRIVIA, so der Projekttitel, bis zum fertigen Spiel. Ausgehend von der Forschungsleitenden Fragestellung, ob es möglich ist, ein Browserspiel mit geringem Aufwand und möglichst günstig, bezogen auf finanzielle Mittel, zu entwickeln wird ein Konzept für ein Browserspiel erarbeitet, dessen Umsetzung eben diese Frage beantworten soll. Im theoretischen Teil dieser Arbeit wurden bereits die Grundlagen von Browserspielen behandelt. Hier werden kurz die Aspekte von TRIVIA zusammengefasst, die dieses Spiel als Onlinespiel qualifizieren.

- TRIVIA ist ein Multiplayerspiel

Vom Spielablauf her ähnelt TRIVIA klassischen Rollenspielen und kann deshalb auch alleine gespielt werden. Es werden jedoch sowohl Kommunikations- als auch Interaktionsmöglichkeiten geboten, um die Spieler dazu zu bewegen zusammen zu spielen. Auch ist die Schwierigkeit mancher Aufgaben darauf ausgelegt, dass diese von mehreren Spielern gemeinsam gelöst werden.

- TRIVIA ist ein Browserspiel

Für TRIVIA ist kein Client nötig, das Spiel läuft einfach im Webbrowser des Spielers ab. Als einzige zusätzliche Anforderung muss das Flashplayer-Plugin in Version 7 oder höher vorhanden sein. TRIVIA ist damit plattformunabhängig und auch in der Erscheinung unabhängig vom jeweiligen Browser, da Flash verwendet wird.

- TRIVIA ist ein Rollenspiel

TRIVIA enthält typische Elemente eines Rollenspiels, auch wenn es keine Kämpfe im herkömmlichen Sinn gibt. TRIVIA ist ein gewaltfreies Spiel und hat auch einen Ansatz in Richtung Lernspiel. Da nicht Monster bekämpft sondern Fragen beantwortet werden müssen, können die Spieler durch TRIVIA auch etwas lernen.

### **3.1 Ziele**

Das Ergebnis der Realisierung von TRIVIA im Rahmen dieser Diplomarbeit ist ein Proof of Concept. Dieser soll eine Beantwortung der Forschungsleitenden Fragestellung ermöglichen und die Fehler in Konzept und Umsetzung aufzeigen.

Das Ergebnis soll eine funktionierende Alpha-Version des Spiels sein, die die wichtigsten Aspekte des Spiels implementiert. Wichtigste Voraussetzung ist die Multiplayer-Fähigkeit sowie die grundlegende Fertigstellung der Spielmechanik. Die fertige Flash-Oberfläche muss zumindest das Spielen ermöglichen, eine besondere grafische Aufbereitung ist für diese Alpha-Version nicht vonnöten. Ebenso muss nicht der gesamte Inhalt des Spiels (Fragen) vorhanden sein, noch muss die Grafik des Spiels einen finalen Status erreichen. Das gesamte Spiel sollte jedoch so programmiert sein, dass eine nachträgliche Erweiterung des Inhalts kein Problem darstellt und auch einzelne Teile des Spiels leicht verbessert oder erweitert werden können.

### **3.2 Spielkonzept**

TRIVIA ist in einer fiktiven Welt angesiedelt, in denen die einzelnen Länder (Level) verschiedene Wissensbereiche repräsentieren. Der Spieler ist eine Kreatur, die auf der Suche nach Wissen und Erfahrung diese Länder durchstreift mit dem Ziel, Meisterschaft in den einzelnen Bereichen zu erreichen. Analog zu klassischen Rollenspielen, in denen man beispielsweise seine Zauberin in Fertigkeiten wie Feuerzaubern schult, lernt man bei TRIVIA Mathematik oder Geschichte um im Spiel weiterzukommen. Der besondere Aspekt dieses Spiels ist jedoch, dass der Spieler, und nicht die Spielfigur, sich dieses Wissen aneignet oder erweitert.

Ziel war es, ein Spiel zu entwerfen, das grundlegende Mechanismen von Multiplayerspielen aufgreift und in einer verhältnismäßig einfachen Umgebung realisiert. Das Spiel an sich basiert grundsätzlich auf dem Brettspiel-Klassiker *Trivial Pursuit* und verwendet somit folgende Elemente:

- Fragen zu verschiedenen Wissensgebieten, die im Spiel durch Schreine und Weise repräsentiert werden
- Hauptfragen, die das Weiterkommen in schwierigere Gebiete oder andere Wissensbereiche ermöglichen, im Spiel repräsentiert durch die Endgegner der Level
- eine abschließende Fragerunde, die einem Spieler den Sieg ermöglicht, in TRIVIA die Meisterschaftsprüfung vor dem Konzil, die umfassendes Wissen in allen Bereichen voraussetzt, jedoch nicht mit einem Sieg (der in Onlinespielen ja nicht möglich sein soll), aber einem ehrwürdigen Titel belohnt wird

Das Spielprinzip ist relativ simpel, aber durch sein Setting und die Spielmechanik doch ungewöhnlich und innovativ. Es unterscheidet sich gegenüber den meisten Flash-Spielen, die das Hauptaugenmerk auf bunte Animationen und schnelle Action legen, aber auch von vielen der Browserspiele die oftmals keine, oder nur sehr rudimentäre grafische Ansicht der Spielwelt bieten. Somit verbindet dieses Spiel, zumindest im Ansatz, die drei Hauptgruppen reiner Onlinespiele. Es ist eine Umsetzung eines klassischen Brettspiels, das durch die Flashoberfläche und die Implementierung eines Teils der Spiellogik in ActionScript somit auch die Gruppe der Flashspiele repräsentiert. Außerdem ist TRIVIA ein Browserspiel, da es nicht auf einen Client angewiesen ist, und besitzt auch Aspekte einer Persistent World, da alle Spieler in einer gemeinsamen Welt agieren, die unabhängig vom Beisein eines einzelnen Spielers ist.

## Technischer Aufbau

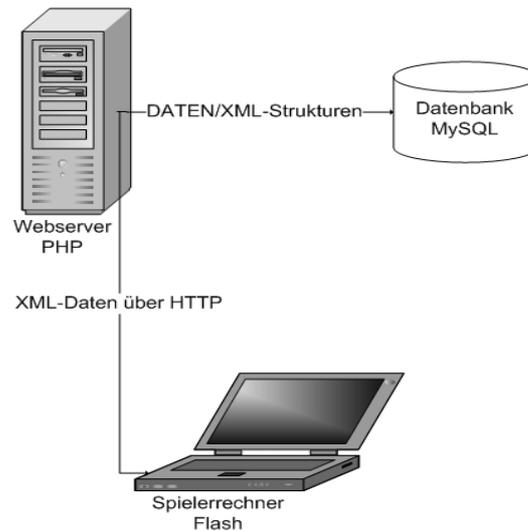
TRIVIA besteht aus einer Flash-Oberfläche, die das Interface und die Spielwelt anzeigt, Benutzereingaben entgegennimmt und die Animationen der Spielfigur ausführt. Werden relevante Benutzereingaben gemacht, werden diese über eine XML-Schnittstelle an den Server gesendet und mittels PHP weiterverarbeitet. So werden zum Beispiel beim Ziehen der Spielfigur die Zielkoordinaten an den Server gesendet, in der Datenbank gespeichert und den anderen Spielern, die sich ebenfalls in diesem Level aufhalten, mitgeteilt, sobald diese ein Update ihres Levels anfordern.

Somit ist die Kommunikation zwischen den einzelnen Teilnehmern asynchron, im Gegensatz zu eventbasierten Systemen, bei denen der Server allen angemeldeten Spielern eine Nachricht sendet, sobald sich irgendetwas geändert hat. In der ersten Ausbaustufe dieses Spiels reicht diese Art der Kommunikation jedoch aus, da außer den Bewegungsdaten nur einer Spielfigur pro Spieler und den Events, wie beantworteten Fragen oder sonstigen Änderungen der Punktestände, kaum Informationen übertragen werden müssen.

Bei Flash-Actionspiele wie Shooter kommt es hingegen auf möglichst geringe Verzögerungszeiten an. Diese setzen meist auf eigene Formate bei der Datenübertragung, da XML hier zuviel Overhead produziert. Noch eine Stufe weiter sind alle clientbasierten Spiele, die zumeist auch auf das TCP/IP-Protokoll zugunsten von UDP verzichten, das die Übertragungsgeschwindigkeit durch die geringeren Latenzzeiten wegen der fehlenden Bestätigungen verringert.

Die Implementierung eines solchen Protokolls würde jedoch den Rahmen dieser Arbeit und auch die Möglichkeiten von Flash zu diesem Zeitpunkt sprengen. Erst mit der Version 9 wird die Implementierung beliebiger Protokolle durch einen binären Socket ermöglicht.

Die Abbildung auf der folgenden Seite zeigt den Kommunikationsaufbau, der für TRIVIA verwendet wird.



**Abbildung 3.1** Client-Server-Schema das für *TRIVIA* verwendet wird

## Spielablauf

Zuerst loggt man sich auf der Webseite von TRIVIA mit seinen Benutzerdaten ein, oder erstellt einen neuen Account. Man steigt direkt in den aktuellen Level ein, die Spielfigur wird an der letzbekanntesten Position platziert. Außerdem werden alle anderen Spieler, die sich im selben Level befinden, an ihrer jeweiligen Position angezeigt.

Die Aufgabe liegt darin das Portal zum jeweils nächsten Level frei zu schalten. Dieses wird jedoch erst aktiviert, wenn eine vorgegebene Anzahl von Fragen innerhalb dieses Levels richtig beantwortet wurde. Diese Fragen sind grundsätzlich in zwei Gruppen aufgeteilt:

- Multiple-Choice-Fragen: es gibt zwei oder mehr Antworten, die durch Klicken ausgewählt werden
- Fragen mit einer Antwort zum eingeben: hierbei handelt es sich entweder um Rechenaufgaben oder Fragen, die eine Jahreszahl oder einen bestimmten Zahlenwert zum Ergebnis haben

Grund für die Aufteilung ist der, dass es äußerst schwierig ist, bei Antworten die aus Wörtern bestehen, die Richtigkeit zu überprüfen. Schon einfache Rechtschreibfehler, vertauschte Buchstaben oder die Eingabe eines Synonyms

führen dazu, dass die Frage als falsch beantwortet gewertet wird. Dies dürfte dem Spielspaß eher abträglich sein.

Fragen können auch über ein Zeitlimit verfügen. Dies soll verhindern, dass die Spieler zur Beantwortung eine Suchmaschine heranziehen.

Hat man eine pro Level festgelegte Anzahl an Fragen richtig beantwortet, wird das Portal aktiviert, und man kann je nach Status des Portals eine Reihe Fragen beantworten, um in den nächsten Level zu gelangen.

Manchmal erhält man bei der Beantwortung einer Frage ein spezielles Hilfsmittel, das im weiteren Spielverlauf verwendet werden kann. So gibt es verschiedene Gegenstände, die einem mehr Zeit geben, einen Taschenrechner, der einem bei mathematischen Aufgaben hilft, einen Notizblock, der bereits beantwortete Fragen speichert oder etwa einen 50:50-Joker, der bei Multiple-Choice-Fragen eingesetzt werden kann. Ob und wann man diese Gegenstände einsetzen kann hängt von den Fragen ab, da für jede gespeichert ist, welche Hilfsmittel verwendet werden dürfen. Diese Gegenstände können auch mit anderen Spielern gehandelt werden. Das ist dann sinnvoll wenn man ein Gebiet bereits durchgespielt hat und beispielsweise noch einige 50:50-Joker für Geschichte übrig hat. Dann kann man diese möglicherweise mit einem anderen Spieler tauschen, der einige für Biologie zuviel hat.

Die soziale Komponente spielt bei den meisten MMOGs eine große Rolle, so auch bei TRIVIA. So können sich mehrere Spieler zu einer Gruppe zusammenschließen, um Fragen gemeinsam zu beantworten oder ein Portal zu öffnen. Auch ein primitives Handelssystem ist vorgesehen, bei dem Gegenstände und auch beantwortete Fragen, wenn man über ein Notizbuch verfügt, getauscht werden können. Die Kommunikation wird über zweierlei Mechanismen ermöglicht. Auf der einen Seite können Nachrichten direkt über die Chatbox abgesetzt werden, diese erscheinen dann mit dem Namen des Spielers in der linken oberen Ecke des Spielfensters und sind für alle Spieler, die sich im selben Level befinden, sichtbar. Auf der anderen Seite kann ein Spieler direkt kontaktiert werden, indem man ihn anklickt und ein Chatfenster öffnet, das ein Vier-Augen-Gespräch ermöglicht.

Die für das Beenden eines Levels nötigen Fragen erhält man, indem man sich auf Felder mit Schreinen bewegt. Je nach aktuellem Level erhält man nun eine Frage aus einem bestimmten Wissensgebiet. Kann man diese beantworten, werden einem Wissenspunkte gutgeschrieben und die Frage im Notizblock, falls vorhanden, gespeichert. Wird die Frage hingegen falsch beantwortet, muss ein anderer Schrein aufgesucht werden. Sind alle Schreine in einem Level aufgebraucht, kann man entweder warten, bis neue Schreine entstehen, oder aber andere Spieler um Fragen bitten. Jeder Spieler mit einem Notizblock kann anderen Spielern, bereits von ihm gelöste, Fragen stellen. Diese wirken dann für den Befragten wie Fragen, die er von einem Schrein erhält. So ist es immer möglich, auch mit vielen falsch beantworteten Fragen, mit der Zeit das Portal zu aktivieren.

Portale verfügen über einen endlichen Fragenpool, der unabhängig von dem des restlichen Levels ist. Werden von einem oder mehreren Spielern nun alle Fragen des Portals beantwortet, öffnet es sich und gibt den Weg in den nächsten Level frei. Für kurze Zeit ist das Portal somit für jedermann, der die Fragengrenze des Levels erreicht hat, quasi gratis zu durchschreiten. Mit der Zeit regeneriert sich das Portal jedoch wieder und es sind wieder mehr Fragen zu beantworten um weiterzukommen.

Um Meisterschaft in einem Wissensgebiet zu erreichen, müssen die Fragen des Weisen in dem speziellen Gebiet beantwortet werden. Dieser erscheint, wenn die Fragenquote des höchsten Levels dieses Gebiets erreicht wird. Um den Konkurrenzkampf der Spieler untereinander voranzutreiben gibt es ab einer gewissen Wissensstufe, den Titel „Meister des [Wissensgebiet]“, der demjenigen erteilt wird, der aktuell die meisten Punkte in diesem Gebiet hat. Dieser Titel geht an einen anderen Spieler über, sobald dieser mehr Punkte erreicht.

Um zu sehen, wo man gegenüber den anderen Spielern steht, gibt es auf der Webseite eine umfangreiche und im Spiel eine kleinere Highscoreliste. Die Liste im Spiel zeigt nur den eigenen Punktestand im Vergleich mit den beiden Plätzen über und unter dem Spieler. Dies soll motivieren, durch Beantworten einiger Fragen noch schnell einen Platz herauszuholen.

Für das Spiel wird eine kleine Webseite eingerichtet, die kurze Informationen über das Spiel und den Anmeldevorgang beherbergt. Bis zur Alpha-Version gibt es noch kein umfangreiches Portal mit Forum, dies kann später jedoch nachgerüstet werden. Für Administratoren steht auch ein spezieller Login zur Verfügung um auf das Administrationstool zugreifen zu können.

Um neue Inhalte, Fragen und Level, hinzuzufügen, wird ein Administrationstool entwickelt, in dem Fragen hinzugefügt und bearbeitet werden können. In diesem Bereich können die Eigenschaften der Fragen, wie Typ, Antwortmöglichkeiten (bei Multiple-Choice-Fragen), erlaubte spielinterne Hilfsmittel, Zeitlimits, etc. eingestellt werden. Bei Fragen muss auch ein Schwierigkeitsgrad voreingestellt werden, der sich jedoch mit der Verwendung im Spiel ändert, je nachdem wie viele Spieler diese Frage falsch oder richtig beantworten. Dieses Verhältnis hebt oder senkt den Schwierigkeitsgrad der Frage, sodass auch eine vom Administrator als schwierig empfunden und eingestufte Frage später zu Beginn des Spiels auftauchen kann, wenn viele Spieler diese auf Anhieb richtig beantworten.

Außerdem können neue Level mit folgenden Eigenschaften angelegt werden: welches Wissensgebiet deckt der Level ab, wie viele Punkte benötigt der Spieler um diesen Level zu betreten, die nötige Fragenquote um das Portal zu aktivieren, die Anzahl der Schreine, verschiedene Zeitspannen (Reaktivierung von Schreinen und dem Portal, etc.) und die Stufe des Levels innerhalb des Spielabschnitts.

Die Kategorien (Wissensgebiete) können ebenfalls online verwaltet werden. Diese sind als Baum angelegt, sodass es Übergebiete wie beispielsweise Geschichte und Biologie gibt und auch Spezialgebiete wie Antike oder Gentechnik. Fragen können übrigens auch mehreren Wissensgebieten zugeordnet werden, falls dies nötig ist.

In der Administration gibt es hier eine rudimentäre Benutzerverwaltung, um Administratoren anzulegen und User zu sperren, falls diese gegen die Spielbedingungen verstoßen haben.

Neue Grafiksets für die Spielwelt müssen in der ersten Version dieses Spiels direkt im Flash-Client eingebunden werden. Da dieser sich auf dem Server und

damit unter Zugriff des Administrators befindet, stellt dies weiters kein Problem dar. Im Gegensatz zu Spielen mit zu installierendem Client entfällt die Implementierung eines Update-Mechanismus.

### **3.3 Umsetzung**

Für die Umsetzung des Spiels wurde ein zeitlicher Rahmen von ungefähr einem Monat bzw. 160 Stunden gesetzt. Diese zeitliche Beschränkung dient dazu, die tatsächliche Realisierungsmöglichkeit innerhalb eines kurzen Zeitraums zu überprüfen. 160 Stunden entsprechen entweder einem Monat bei Vollzeit-Beschäftigung, oder aber ungefähr zwei bis drei Monaten, wenn man neben Arbeit oder Studium ein solches Projekt als Hobby betreibt. Diese Zeit beinhaltet die komplette Entwicklung des Spiels von der Konzeptionsphase bis zur fertigen Alpha-Version, mit der ein erster Testlauf mit Probanden durchgeführt werden kann.

Bezüglich der technischen Basis wurde PHP als serverseitige Scriptsprache und MySQL als Datenbank ausgewählt. Grund dafür ist die freie Verfügbarkeit und der für dieses Projekt ausreichende Funktionsumfang. Für die Präsentation des Spiels wird Flash verwendet, da es ansprechende Grafiken ermöglicht und auch Animationen die mit einfachen HTML nicht möglich wären. Dazu kommt, dass bei der heutigen Verbreitung von Breitband-Internetanschlüssen, die Dateigrößen und Ladezeiten der Flashfilme kaum ins Gewicht fallen und in Zukunft außerdem viele Browser Spiele auf Flash (vielleicht sogar auf Shockwave) basieren werden und dies somit gewissermaßen einen innovativen Aspekt der Entwicklung darstellt.

Im Folgenden werden die einzelnen Phasen der Entwicklung von TRIVIA beschrieben.

## Konzeptionsphase

Aufgabe dieser Phase ist die Überführung der Spielidee in ein programmierbares Konzept. Es muss überlegt werden, welche Vorgänge mittels PHP auf dem Server und welche in Flash behandelt werden, wie die Datenstrukturen aussehen, die in der Datenbank gespeichert werden und welche Schnittstellen zwischen PHP und Flash benötigt werden, bzw. wie diese auszusehen haben. Außerdem gilt es in dieser Phase festzulegen, welche Arten von Level es geben wird, welche Gegenstände im Spiel vorkommen und wie sich die Spielfigur durch die Level bewegen soll. Es müssen auch Screendesigns der einzelnen Oberflächen angelegt und die Spielmechanik definiert werden. In der Konzeptionsphase für TRIVIA lag das Hauptaugenmerk auf der schnellen Umsetzbarkeit und einer flexiblen Gestaltung, um den Aufwand für nachträgliche Erweiterungen zu minimieren. Daraus ergab sich auch die Überlegung, zuerst einen Editor zu programmieren mit dem später, analog zu den Szenarioeditoren gängiger Strategiespiele, Level einfach aufgebaut und bearbeitet werden können. Aus dem Konzept ergab sich weiters eine strikte Trennung der Bereiche für die beiden verwendeten Scriptsprachen, PHP und ActionScript. Die serverseitige Scriptsprache wird notwendigerweise für die Datenbankoperationen verwendet, außerdem für die dynamische Erstellung der Startseite, der Benutzerverwaltung und zum Behandeln des Login-Vorgangs. Der Login-Vorgang führt je nach Berechtigung und Auswahl zu einer Seite die Flash entweder als Spiel oder als Editor aufruft. Hierbei handelt es sich um denselben Flashfilm, da sich Spiel und Editor nicht besonders stark unterscheiden.

Außerdem wird in PHP ein kleines Administrationstool entwickelt, indem einfach neue Fragen angelegt werden können. Das Anlegen von neuen Spielern geschieht ebenfalls auf der Serverseite.

Die gesamte restliche Spiellogik, wie Anzeigen von Fragen, Überprüfen der Antworten, Aktualisieren der Spielereigenschaften sowie das Ändern der Spielerposition wird komplett mittels ActionScript umgesetzt.

Ein weiteres Ergebnis der Konzeptphase war die Entscheidung, vollständig objektorientiert zu programmieren. Sowohl Flash mit ActionScript 2.0 als auch PHP in Version 5 bieten die Möglichkeiten zur Objektorientierung die für dieses

Projekt notwendig sind. Obwohl die Anwendung von Objektorientierung auf den ersten Blick länger dauert und eine umfangreichere Konzeption voraussetzt, bietet diese Art der Entwicklung auf längere Sicht Vorteile, durch den klareren Code und die einfachere Wart- und Erweiterbarkeit.

## Umsetzungsphase

Bei der Spielentwicklung bietet es sich grundsätzlich an, zuerst einen Editor zu bauen, da das Erstellen der einzelnen Levels später um einiges leichter fällt. Die Spielwelt von TRIVIA ist aus Tiles aufgebaut. Diese sind rautenförmig und erzeugen eine isometrische Ansicht mit einem Pseudo-3D-Effekt, wie man ihn von Spielen wie *Command and Conquer* [httpC&C] oder *Diablo* [httpDiablo] gewöhnt ist. Die Beschreibung des Levelaufbaus wird aus einem XML-String extrahiert, wobei jeder Tile durch einen eigenen Knoten repräsentiert wird. Würde man nun zur Levelerstellung die einzelnen XML-Dateien direkt editieren, wäre dies weitaus umständlicher und fehleranfälliger, als wenn man den Level auf dem Bildschirm sieht und direkt Elemente hinzufügen und entfernen kann.

Mittels dieses Editors können neue Levels erstellt, oder bestehende intuitiv bearbeitet werden. Die einzelnen Grafiken liegen in einzelnen Sets vor, sodass man die Optik eines Levels einfach durch Austauschen des Grafiksets verändern kann (z.B. von Biologie mit Flora und Fauna auf Geschichte mit Ruinen und Schlachtfeldern).

Um überhaupt erst einen Editor entwickeln zu können, muss eine Engine vorhanden sein. Diese übernimmt den Levelaufbau in einer bestimmten Beschreibung und generiert daraus die Spieloberfläche. Im Folgenden soll nun ein Blick auf die beiden sichtbaren Teile der Oberfläche, dem Spielfeld und das Interface geworfen werden.

Das Spielfeld wird, wie oben erwähnt, aus so genannten Tiles aufgebaut. Dabei handelt es sich im Grunde um kleine Elemente (Quadrate oder Rechtecke), in die das Spielfeld aufgeteilt ist. Da sehr viele Bereiche des Spielfeldes gleich sind, können an diesen Stellen jeweils dieselben Tiles verwendet werden, wodurch die Dateigröße und somit auch die Ladezeiten drastisch reduziert werden.

Diese Art des Spielaufbaus hat bereits eine lange Tradition in der Geschichte der Computerspiele. In Zeiten in denen die Ressourcen bezüglich Prozessorleistung und Arbeitsspeicher noch knapp waren, bot diese Technik eine gute Möglichkeit das Aussehen und die Performance von Spielen zu verbessern. Abgesehen davon lassen sich in Spielen auf Tile-Basis viele Dinge einfacher implementieren. Die auch bei diesem Spiel verwendete isometrische Ansicht mit dazugehöriger Tiefensortierung (depth sorting), bei der bewegte Objekte in den Vorder- bzw. Hintergrund treten können, oder auch Wegfindungsalgorithmen lassen sich so sehr einfach umsetzen.

Hauptargument für Tiles ist aber auf jeden Fall die Performance. Sowohl die geringe Dateigröße, als auch die Möglichkeit, bei Änderungen der Spielwelt nur den geänderten Bereich neu zeichnen zu müssen tragen dazu bei. Alle diese Aspekte von Tiles können in Flash verwendet werden, aus Geschwindigkeitsgründen sollten jedoch für die Tiles keine flasheigenen Vektorgrafiken sondern Bitmaps zum Einsatz kommen. Während die Vektorgrafiken vom Flashplayer in jedem Frame neu berechnet und gezeichnet werden müssen, werden Bitmaps nur einmal gerendert. Ab Flash 8, das auch für dieses Spiel zum Einsatz kommt, steht auch die Möglichkeit zur Verfügung, flasheigene Clips wie Bitmaps behandeln zu lassen. Diese werden dann ebenfalls vorgerendert und bieten eine vergleichbare Performance.

Zur Erstellung von Spielen auf Tile-Basis bietet sich die Lektüre des sehr guten Tutorials von Tonu „Tonypa“ Paldra, zu finden auf dem Flash-Spieleportal gotoAndPlay.it [<http://GoTo>], an. Dieses behandelt alle wichtigen Themen von Steuerung über Programmierung von Gegnern bis hin zu Wegfindung und isometrischer Ansicht.

Um Aufbau und Verwaltung des jeweiligen Levels kümmert sich bei TRIVIA die Klasse Level. Diese bekommt den Level in Form eines XML-Strings übergeben und baut daraus das Spielfeld auf. Jeder Tile wird dabei von einem Objekt der gleichnamigen Klasse repräsentiert, das sich um die Eigenschaften und das Aussehen kümmert. Außerdem beinhalten diese Objekte die Eventhandler der Tiles. Wird zum Beispiel ein Tile angeklickt, wird eine Funktion aufgerufen, die das Tile-Objekt als Parameter übernimmt und die Spielfigur veranlasst, sich an

die Position des angeklickten Tiles zu bewegen. Die Methoden zur Bewegungen der Spielfigur werden in einem Objekt der Klasse Player gekapselt, die sich auch um die Verwaltung des Punktestandes und aller anderen Spielereigenschaften kümmert.

Kommunikation zwischen Flash und PHP ist in verschiedenen Situationen notwendig. So müssen die einzelnen Elemente (Level und Spieler) geladen werden, Änderungen der Spielerposition oder –werte müssen auf dem Server aktualisiert werden. In einem gewissen Intervall müssen Aktualisierungen der anderen Spieler bzw. des Levels abgefragt werden.

Als Format kommt XML zum Einsatz. Auf Flashseite wird ein XML-Objekt erstellt, dessen Inhalt als String an das jeweilige Serverscript geschickt wird. PHP holt sich die Daten aus dem Feld „HTTP\_RAW\_POST\_DATA“ des `$_GLOBALS-Arrays`, das den XML-String unformatiert enthält. Die übergebenen Daten werden entweder in die Datenbank gespeichert oder geparkt und verarbeitet. Danach wird von PHP eine XML-Seite erstellt, die wieder an Flash zurückgeliefert wird. Diese Seite enthält entweder angeforderte Daten, oder Statusmeldungen über den (Miss-)Erfolg der ausgeführten Operation.

Alle für das Spiel benötigten Elemente (Tiles, Spielfiguren, Fragen, Kategorien und Antworten) werden im Flash durch Objekte dargestellt, an das PHP als XML-Strukturen übermittelt und auch als solche in der Datenbank gespeichert.

Der Aufbau der XMLs sieht folgendermaßen aus:

```
<level id="1" title="testlevel">
  <tile ...
  ...
</level>
```

Diese XML-Struktur beinhaltet ID und Name des Levels, sowie sämtliche Tiles aus denen derselbe aufgebaut ist.

```
<tile id="421" x="10" y="20" qid="2" gset="desert" graphic="2" />
```

Aus dieser Struktur können auch die relevanten Attribute herausgelesen werden. Die *id* ist eine laufende Nummer, die einzigartig für jedes Tile innerhalb eines Levels ist. *x* und *y* sind Positionswerte. Diese sind jedoch nicht in Pixeln angegeben, sondern entsprechen der Position innerhalb eines zwei-dimensionalen Arrays (*x* = Spalte, *y* = Reihe). Die Positionen werden beim Aufbau des Levels aus der Größe der Tiles berechnet. Ist diesem Feld eine Frage zugeordnet so hat das Attribut *qid* den Wert der ID der entsprechenden Frage. *gset* und *graphic* definieren das Aussehen des Tiles. Im Spiel sind verschiedene Grafik-Sets vorgesehen (hier z.B. „desert“). Dabei handelt es sich um Movieclips die in den jeweiligen Frames unterschiedliche Grafiken beinhalten. Beim Aufbau dieses Beispiel-Feldes würde der Movieclip „desert“ auf die Bühne gebracht und der zweite Frame (*graphic* = 2) angezeigt werden. Das Ergebnis ist ein gelbes Feld mit einer Palme an der Position  $(y - x) * \text{Tilebreite} / (y + x) * \text{Tilebreite}/2$ .

```
<question id="12" type="mc" category="3" timelimit="5000" points="25">
  <text>Wann erreichte Österreich die beste WM-Platzierung</text>
  <answers>
    <answer ...
    ...
  </answers>
</question>
```

Betrifft die Spielfigur ein Fragenfeld, wird die entsprechende Frage, anhand der ID, die im Tile gespeichert ist, vom Server geladen. Darin ist gespeichert, zu welcher Kategorie diese Frage gehört, ob es ein Zeitlimit gibt und wie viel Punkte bei richtiger Beantwortung dem Spielerkonto gutgeschrieben werden. Das Typ-Attribut gibt an, ob es sich um eine Multiple-Choice-Frage handelt, oder um eine Frage, bei der die Antwort in ein Textfeld eingegeben werden muss (*type*="sa"). Die restliche Struktur unterscheidet sich bei diesen beiden Fragentypen nicht, mit der Ausnahme, dass Fragen des zweiten Typs nur einen Antwort-Tag besitzen.

```
<category id="3" title="Geschichte" color="0x33BB22"/>
```

Kategorien dienen dazu, mehrere Fragen die ein ähnliches Gebiet abdecken, zusammenzufassen. Neben der obligatorischen ID und dem Namen der Kategorie, enthält das XML auch noch die Farbe der Kategorie, die bei der Anzeige der Fragen verwendet wird.

```
<answer id="15" correct="true">1954</answer>
```

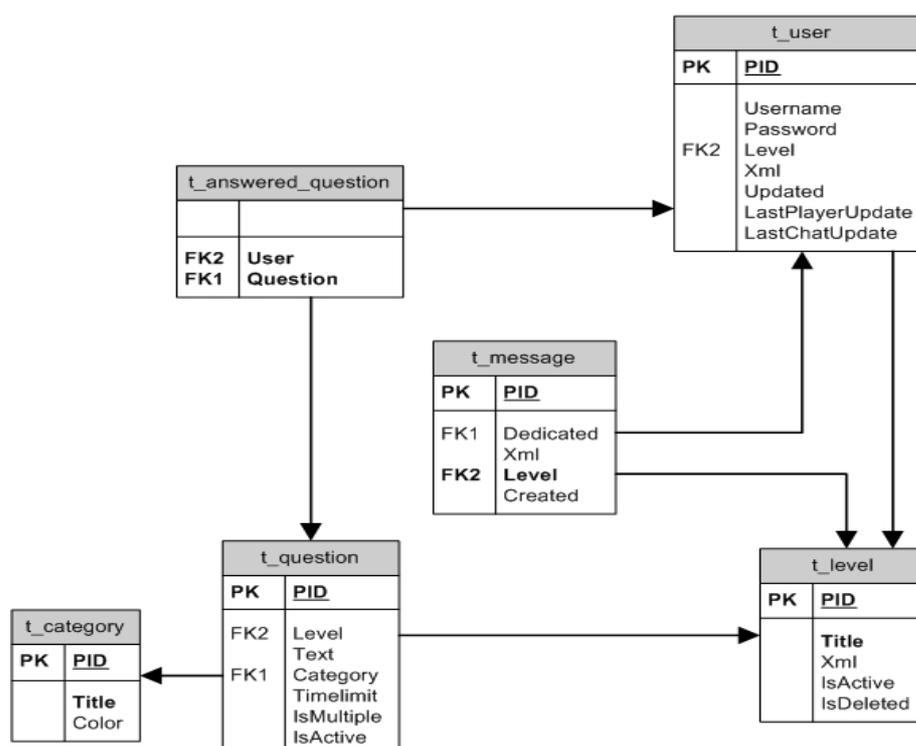
Antworten verfügen wieder über eine ID, außerdem über ein Attribut, dass anzeigt, ob diese Antwort die richtige ist oder nicht. Die Antwort selbst steht zwischen den XML-Tags.

```
<player id="15" name="pk" score="2423" level="1" x="122" y="155">
  <pool>
    <question id="4" />
    <question id="6" />
  </pool>
  <inventory>
    <item id="2" name="Taschenrechner" available="1" />
    <item id="4" name="50:50-Joker" available="3" />
  </inventory>
</player>
```

Neben ID und Name, sind hier noch der Punktestand, die ID des aktuellen Levels und die Position der Spielfigur innerhalb dieses Levels gespeichert. Außerdem wird gespeichert, welche Fragen der Spieler bereits richtig beantwortet hat. Zum einen geschieht dies, damit der Spieler nicht eine Frage ein zweites Mal gestellt bekommt. Zum anderen, da es auch vorgesehen ist, später in der Entwicklung den Handel mit beantworteten Fragen zu erlauben. Zusätzlich verfügt der Spieler über ein Inventar, welches seine gesammelten Gegenstände aufnimmt. Diese werden mit ID und der Anzahl verfügbaren Nutzungen gespeichert. Als Gegenstände stehen beispielsweise ein 50:50-Joker oder ein Taschenrechner zur Verfügung.

Für die Gestaltung der Interfaces kommen Slides zum Einsatz. Dabei handelt es sich eigentlich um Präsentationsvorlagen in Flash, die für Web-Präsentationen gedacht sind. Diese sind hierarchisch aufgebaut und können einzelnen an- und ausgeschaltet werden. So kann man beispielsweise den Hintergrund des Spiels und Elemente die immer sichtbar sind auf den Hauptslide legen und die restlichen Inhalte bei Bedarf ein- und ausblenden.

Die Vorteile von Slides liegen in der einfachen Verwendung. Man erstellt zuerst einen Slide, benennt die einzelnen Seiten, damit man später darauf zugreifen kann und kompiliert die fertige Präsentation in eine SWF-Datei. Diese kann dann innerhalb des Spiels geladen und verwendet werden. Man kann in Slides auch leere Movieclips als Platzhalter verwenden, um an deren Positionen beim Anzeigen dynamisch generierten Inhalt, wie zum Beispiel Schaltflächen, einzufügen.



**Abb. 3.2** Datenbank-Schema von TRIVIA

Die Interaktivität zwischen den Spielern wird über asynchrone Kommunikation realisiert. Wenn ein Spieler eine Aktion setzt, beispielsweise einen Text mittels Chatfenster absetzt, oder einen anderen Spieler anklickt um einen Tausch durchzuführen, wird die jeweilige Aktion an den Server gesendet. Dieser

speichert die Aktion in die Datenbank (DB-Struktur siehe Abb. 3.2, S.52) und setzt ein entsprechendes Flag für den Updatestatus des Levels. Alle paar hundert Millisekunden, fragt das Spiel den Server ob Aktionen oder Ereignisse anstehen, bekommt diese als Ergebnis-Set zurück und zeigt sie entsprechend an.

Der Flash-Part der Applikation ist grundsätzlich objektorientiert angelegt. Die Spiellogik findet sich im ersten Frame der Hauptzeitleiste. Zu Beginn werden globale Variablen befüllt und die Initialisierungsscripte (für Spiel oder Editor – je nach Einstellung) aufgerufen. Alle Aktionen im Spiel werden über Eventhandler abgefangen. Zusätzlich gibt es noch ein Script das bei jedem Framewechsel (*onEnterFrame*) ausgeführt wird. Dieses ruft diverse Funktionen anhand verschiedener Spielstatus auf. Wenn sich beispielsweise ein Spieler in Bewegung befindet, wird die *Move()*-Methode der Spielerklasse aufgerufen, außer die Spielfigur hat ihr Ziel bereits erreicht, dann wird eine entsprechende Aktion ausgeführt.

Auf der Serverseite werden Klassen nur für das Datenbank-Handling und die XML-Verarbeitung verwendet. Für jede Anfrage vom Flash-Client steht ein PHP-Script zur Verfügung, das die gesendeten GET-Parameter und/oder XML-Daten verarbeitet und das gewünschte Resultat als XML zurückliefert. Im Fehlerfall wird ein XML-String mit der Error-Message übermittelt.

Die folgenden Seiten zeigen eine Übersicht über die verwendeten ActionScript- und PHP-Klassen für TRIVIA sowie Screenshots aus der ersten spielbaren Version.

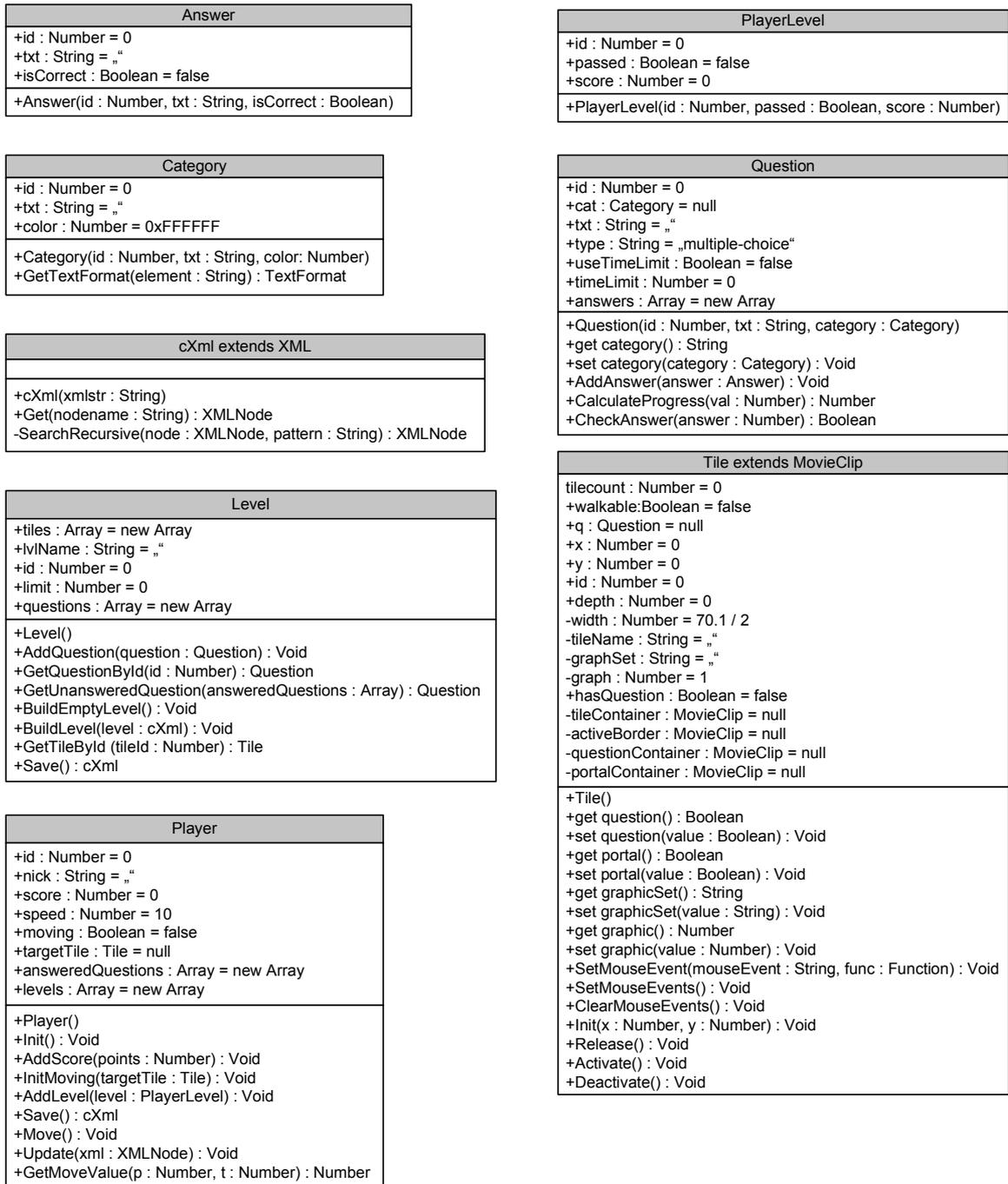


Abbildung 2.3 ActionScript-Klassendiagramm für TRIVIA

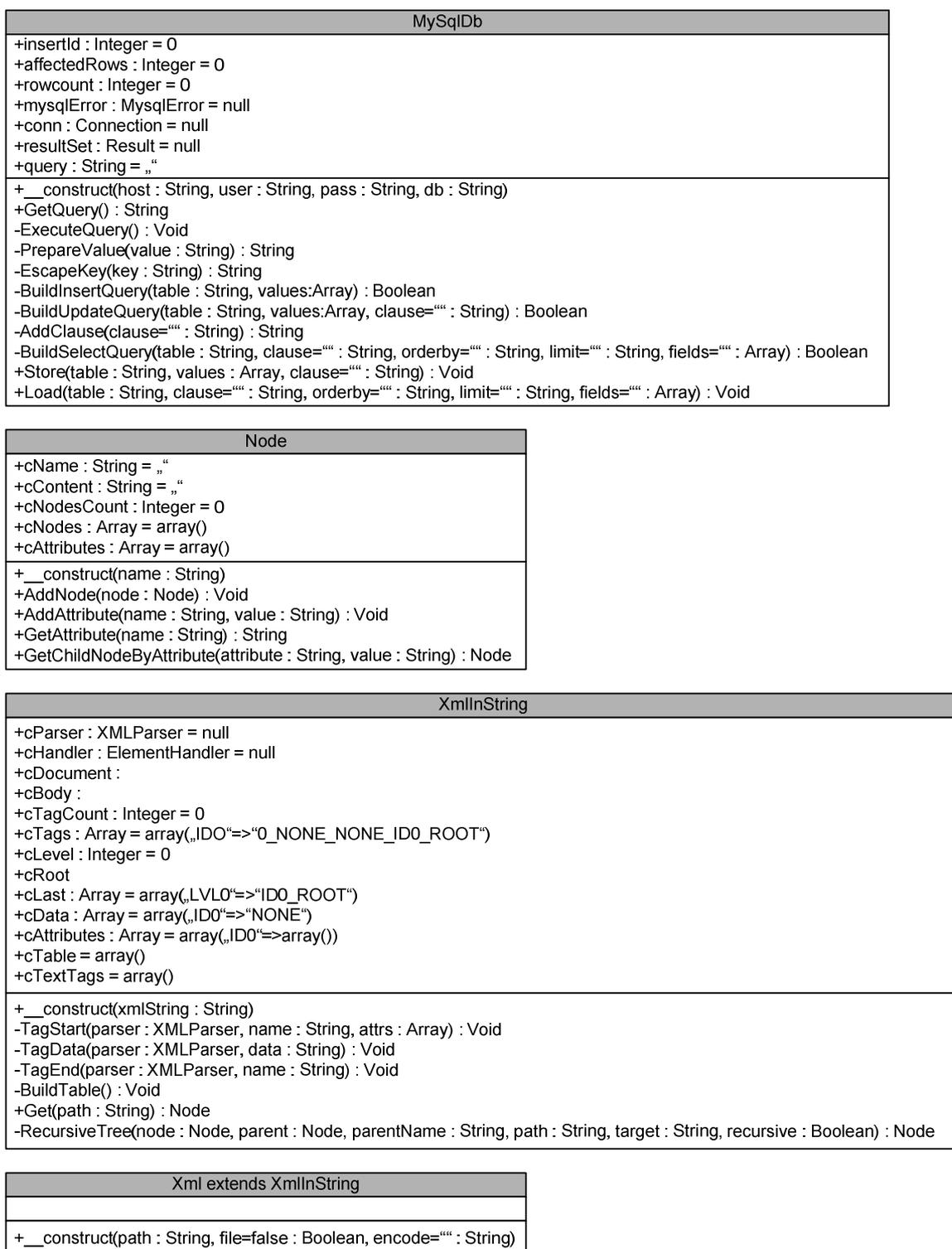


Abbildung 3.4 PHP-Klassendiagramm für TRIVIA



Abbildung 3.5 Screenshot einer ersten spielbaren Version von TRIVIA (Entwicklungsgrafik)

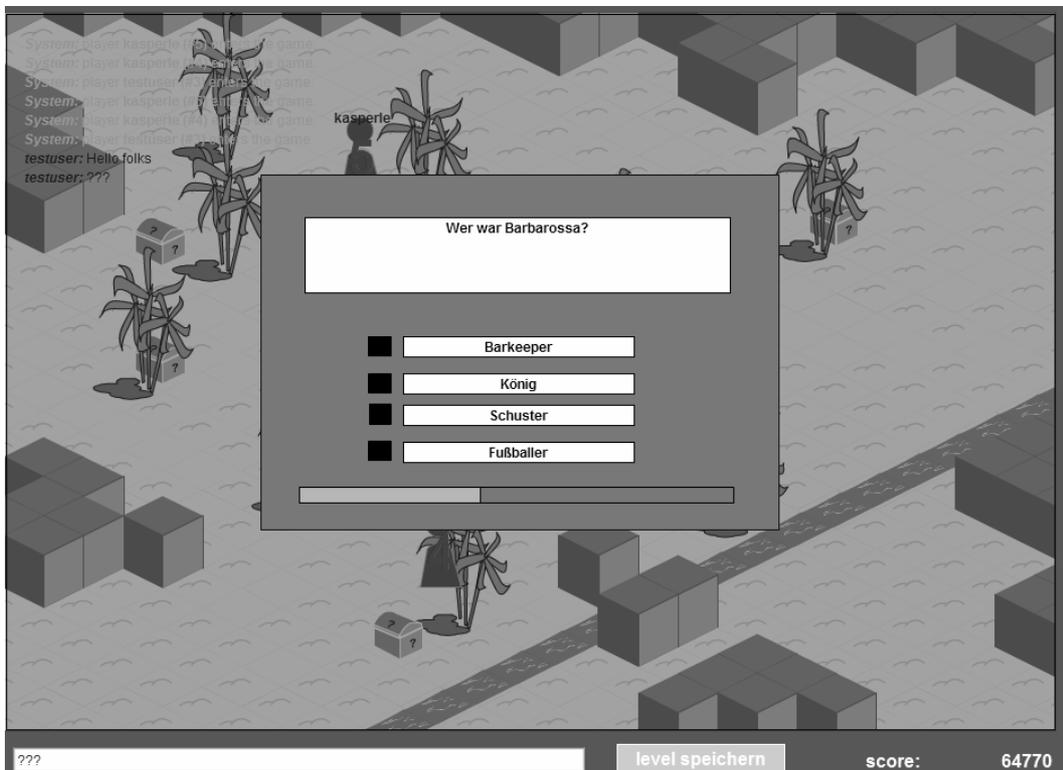


Abbildung 3.6 Screenshot einer ersten spielbaren Version von TRIVIA mit angezeigter Frage

## 4. Ergebnisse

### 4.1 Beantwortung der Fragestellung

Ob sich ein Browserspiel nun kostengünstig und mit verhältnismäßig geringem Aufwand entwickeln lässt, ist nach den Resultaten mit TRIVIA in hohem Grad positiv zu beantworten.

Die geringen Kosten sind einfach zu erzielen, da alle benötigten Werkzeuge, mit Ausnahme von Flash, als Freeware oder Open Source verfügbar sind und durch die Eigenentwicklung bzw. mit Unterstützung freiwilliger Mitarbeiter keinerlei Personalkosten anfallen. Die laufenden Kosten nach der Fertigstellung für Server und Traffic lassen sich über das Einblenden von Werbung oder den Verkauf von Premium-Accounts decken, erreichen aber bei Browserspielen, deren Spieleranzahlen mit kommerziellen MMOGs nicht konkurrieren können, keine großen Ausmaße. Es lässt sich somit ein Browserspiel-Projekt, als Hobby betrachtet, auch aus eigener Tasche bestreiten ohne den finanziellen Ruin fürchten zu müssen.

Das Schlüsselwort bei den Entwicklungszeiten ist „verhältnismäßig“. Der Aufwand für ein solches Spiel darf keinesfalls unterschätzt werden, im Besonderen der Aufwand für Erweiterungen und Betreuung nach der Fertigstellung kann große Ausmaße annehmen. Im Vergleich zu professionellen Spielen ist der Aufwand jedoch überschaubar und durchaus von einem einzelnen Entwickler zu bewältigen. [LOBER03]

Die für diese Arbeit festgesetzte Entwicklungszeit von 160 Stunden, ermöglichte es, ein Spielkonzept zu entwickeln und dies bis zu einer ersten spielbaren Version umzusetzen. Bis zur Fertigstellung einer Beta-Version mitsamt der kompletten Funktionalität sind geschätzt weitere 160 Stunden nötig. Aber diese Arbeit zeigt somit, dass es durchaus möglich ist, ein Browserspiel auf Flash-Basis als Hobby neben einer regulären Beschäftigung erfolgsversprechend in Angriff zu nehmen.

Nicht zuletzt ist anzumerken, dass es tatsächlich möglich ist, mit geringer Teamgröße und genügend Innovation eine virtuelle Welt zu schaffen, deren

Komplexitätsgrad und Beliebtheit, die vieler kommerzieller Produktionen übersteigt, deren Entwicklungs- und Supportkosten im Bereich von mehreren Millionen Euro liegen.

## ***4.2 Schwierigkeiten bei der Erstellung***

Im Zuge der Erstellung von TRIVIA traten einige Stolpersteine zu Tage, die die Entwicklung des Spiels verzögerten. Wie zu erwarten gewesen wäre, rächte sich die Entscheidung mit ActionScript eine Programmiersprache zu verwenden, die dem Entwickler nicht ausreichend geläufig war. Einfachste Programmierkenntnisse reichen für so ein Projekt nicht aus, es erfordert viel mehr eine tiefere Einarbeitung in die Thematik und die verwendeten Werkzeuge.

Bei der Verwendung von ActionScript stellte sich vor allem das Zusammenspiel von MovieClips mit Klassen als Herausforderung dar. Um dem objektorientierten Ansatz zu entsprechen, wurden die MovieClips, die die einzelnen Spielelemente repräsentieren, mit ActionScript-Klassen verknüpft, um die jeweilige Funktionalität (z.B. Bewegen der Spielfigur) in Methoden kapseln zu können. Dabei ergaben sich einige Schwierigkeiten bezüglich der Geltungsräume von Variablen, da beispielsweise nicht von der Klasse auf die Eigenschaften des MovieClips zugegriffen werden konnte. Obwohl sich dieser Umstand dadurch beheben ließ, indem man die jeweilige Klasse um die Klasse MovieClip erweiterte, blieben dennoch alle zusätzlichen Attribute des Clips, wie statisch eingebundene MCs unerreichbar. Erst der komplette dynamische Zusammenbau der MovieClips ermöglichte der Klasse alle nötigen Funktionen auszuführen.

Auch die Kommunikation mit PHP über XML erwies sich als schwieriger als erwartet, da der Aufbau von XML-Objekten in Flash nicht besonders einfach ist. Aber nicht nur der Aufbau, auch die zu verwendenden Header verursachten zu Beginn Probleme. Am Wichtigsten war allerdings die Erkenntnis, dass sämtliche Attribute einer XML-Node vom Typ her Strings sind und Flash trotz Verwendung von Typangaben bei Variablen, teilweise willkürliche Umwandlungen vornahm,

was zu unvorhergesehenen Ergebnissen führte. Nur durch die konsequente Verwendung von type casting konnte dieses Problem behoben werden.

Eine weitere Fehlerquelle war die Asynchronität der Verarbeitung der Flash-Scripte. Beispielsweise stehen MovieClips die in das Spiel geladen werden, nicht sofort zur Verfügung sondern müssen über Eventhandler, die beim Fertigladen der Clips angestoßen werden, behandelt werden.

### **4.3 Spielstatus**

Das Spiel enthält zwei Level die vollständig gespielt werden können und einen Eindruck der Spielmechanik vermitteln. Über den Editor können einfach neue Level erstellt werden und über das Administrationstool kann problemlos neuer Inhalt in Form neuer Fragen hinzugefügt werden. Diese Unfertigkeit, obwohl in diesem Fall durch den Zeitdruck provoziert, ist generell symptomatisch für Browser Spiele, da diese Art Spiel ständig weiterentwickelt wird. Da viele dieser Spiele in Runden gespielt werden, die einige Monate dauern, kann mit jeder neuen Runde Feedback aus der Community oder neue Ideen der Entwickler in die bestehende Welt eingebaut, oder aber die Parameter der Spielumgebung modifiziert werden.

Insgesamt befindet sich das Spiel in einer Pre-Alpha-Version, wobei die nächsten Schritte Verbesserung der Performance und eine komplette Implementierung der vom Konzept beschriebenen Spielfunktionen sind. Beispielsweise sind von den Gegenständen her nur der Taschenrechner und der Notizblock verfügbar, sowie die Tauschmöglichkeiten zwischen den Spielern noch nicht vollständig implementiert.

#### **4.4 Erweiterungsmöglichkeiten und Ausblick**

Die Erweiterungsmöglichkeiten beschränken sich nicht nur auf das Hinzufügen neuer Level und Fragen, viel mehr wäre es durchaus möglich zusätzliche Spielmechanik, wie erweiterte Interaktion zwischen den einzelnen Spielern zu implementieren. Schließlich sollte auch die grafische Aufmachung und vor allem die Performance verbessert werden. Die Datenübertragung zwischen Flash und PHP ließe sich über kompaktere XML-Datenmengen verringern, Eine Möglichkeit wäre, beispielsweise die Ladezeit des Level zu verkürzen, indem man nur die Tile-Knoten überträgt die nicht Standardtiles sind.

Von der Weiterentwicklung her ist TRIVIA eine ungewisse Zukunft bestimmt. Da sich das Konzept von den gängigen Spieltypen unterscheidet und auch als Proof of Concept mit Fokus auf die technische Realisierbarkeit entwickelt wurde, ist noch unklar ob es von einer großen Masse an Spielern angenommen werden würde. Das Spiel soll in jedem Fall auf Hobby-Basis zu einer testfähigen Beta-Version weiterentwickelt werden um auch diesen Punkt zu klären.

## 5. Schlusswort

Zieht man die Interviews mit den Entwicklern von Browserspielen auf *gamingfacts* heran, kristallisiert sich eine Vision heraus: reine Offline-Spiele werden mit der Zeit vollkommen verdrängt werden, da Internetzugang jederzeit und überall möglich sein wird und der Spielspass durch menschliche Gegner stark erhöht wird. Browserspiele werden im Vergleich zu clientbasierten Spielen jedoch noch länger ein Nischendasein fristen, da auch das Auge mitspielt und die Unterschiede in der grafischen Qualität noch lange gravierend bleiben werden.

Jedenfalls wird die Entwicklung von Browserspielen mit der Zeit professionalisiert werden, mit allen positiven wie negativen Auswirkungen. Auf der einen Seite wird sich die Stabilität und Präsentation der Spiele aufgrund des höheren Know-hows in den Firmen verbessern, auf der anderen Seite besteht die Gefahr, dass Browserspiele ein ähnliches Schicksal erleiden, wie klassische Computerspiele und zu einem Einheitsbrei verarbeitet werden.

Ein positiver Impuls für die Browserspiel-Szene wäre auch die Entwicklung einer freien Grafikengine, beispielsweise für Strategiespiele, die auf Flash oder Java basierend und leicht veränderbar, vielen die Möglichkeit geben würde, sich mehr auf den Inhalt und das Spielerlebnis zu konzentrieren als auf die ständige Neuerfindung des Rades.

Zu hoffen ist, dass sich die teils große Spieltiefe von heutigen Browserspielen, mit daraus resultierender starker Community, auch auf klassische Spiele übertragen lässt und somit ein wenig mehr Innovation den Markt bereichert.

## Quellenverzeichnis

- [BART96] Bartle, Richard A.: Hearts, Clubs, Diamonds, Spades: Player Who Suit MUDs. <http://www.mud.co.uk/richard/hclds.htm>, 1996 (abgerufen 19. 7. 2006)
- [FRIE02] Friedl, Markus: Online Game Interactivity Theory. Charles River Media, 2002
- [JUD01] Judmaier, Peter: Virtual Societies – Möglichkeiten von Online Community Games am Beispiel Myzel. Diplomarbeit an der TU Wien, 2001
- [KOST97] Koster, Raph: The Laws of Online World Design. <http://www.legendmud.org/raph/gaming/laws.html>, 1997 (abgerufen 19. 7. 2006)
- [LISCH03] Lischka, Konrad: Güter und Gegner – Wirtschaftssysteme in Online-Rollenspielen. In: c't 11/2003
- [LOBER06] Lober, Andreas: Weltenfenster – Browser Spiele werden erwachsen. In: c't 10/2006
- [LOBER03] Lober, Andreas/Sontowski Ingo: Der Feind im Fenster. In: c't 23/2003
- [MERT06] Mertens, Mathias/Meißner, Tobias O.: Wir waren Space Invaders, Geschichten vom Computerspielen. Blumenkamp, 2006
- [MULL03] Mulligan, Jessica/Patrovsky Bridgette: Developing Online Games. New Riders Publishing, 2003
- [SING99] Singhal, Sandeep/Zyda, Michael: Networked Virtual Environments – Design and Implementation. ACM Press/Addison-Wesley Publishing, New York, 1999
- [http8BIT] Slabihoud, Stephan: The Dot Eaters – Die Geschichte der Videospiele. <http://www.8bit-museum.de/>, 26. 6. 2006
- [httpAC] Asheron's Call, <http://ac.turbine.com>, 4. 9. 2006
- [httpAO] Anarchy Online, <http://www.anarchy-online.com>, 4. 9. 2006

- [httpC&C] Command & Conquer, <http://westwood.ea.com>, 4. 9. 2006
- [httpDAoC] Dark Age of Camelot, <http://www.darkageofcamelot.com>, 4. 9. 2006
- [httpDiablo] Diablo, <http://www.blizzard.de/diablo>, 12. 8. 2006
- [httpDDO] Dungeons & Dragons Online, <http://www.ddo.com>, 4. 9. 2006
- [httpEQ] EverQuest, <http://eq.mystics.de>, 12. 8. 2006
- [httpEQ2] EverQuest II, <http://everquest2.station.sony.com>, 12. 8. 2006
- [httpGANG] Gangwars, <http://www.gangwars.de>, 23. 7. 2006
- [httpGF] gamingfacts, <http://www.gamingfacts.de>, 26. 6. 2006
- [httpGoTo] gotoAndPlay() <http://www.gotoandplay.it>, 15. 7. 2006
- [httpGW] Guild Wars, <http://de.guildwars.com>, 4. 9. 2006
- [httpGXY] Galaxy Wars, <http://www.galaxywars.de>, 23. 7. 2006
- [httpM59] Meridian 59, <http://meridian59-deutschland.de>, 4. 9. 2006
- [httpMIT] <http://lcs.www.media.mit.edu/groups/el/projects/spacewar>, 12. 8. 2006
- [httpPLAN] Planetarion, <http://www.planetarion.com>, 23. 7. 2006
- [httpRag] Ragnarok, <http://www.euro-ro.net>, 4. 9. 2006
- [httpUO] Ultima Online, <http://www.uo.com>, 23. 7. 2006
- [httpWIKI] "Turn-based strategy", [http://en.wikipedia.org/wiki/Turn\\_based](http://en.wikipedia.org/wiki/Turn_based), 28. 8. 2006
- [httpWO] Warhammer Online, <http://www.warhammeronline.com>, 3. 9. 2006
- [httpWoC] World of Crime, <http://worldofcrime.de>, 15. 7. 2006
- [httpWoW] World of WarCraft, <http://www.worldofwarcraft.com>, 4. 9. 2006

## Abbildungsverzeichnis

Abbildung 2.1 Nachbildung von Spacewar in Java	11
Abbildung 2.2 Screenshot aus Ultima Online	12
Abbildung 2.3 Screenshot aus dem MMORG EverQuest	14
Abbildung 2.4 Screenshot aus Dark Age of Camelot	14
Abbildung 2.5 Verteilung von Spielertypen bezüglich Interessensphären	31
Abbildung 2.6 Abhängigkeiten der Spielertypen untereinander	32
Abbildung 3.1 Client-Server-Schema das für <i>TRIVIA</i> verwendet wird	41
Abbildung 3.2 Datenbank-Schema für <i>TRIVIA</i>	52
Abbildung 3.3 ActionScript-Klassendiagramm für <i>TRIVIA</i>	54
Abbildung 3.4 PHP-Klassendiagramm für <i>TRIVIA</i>	55
Abbildung 3.5 Screenshot einer ersten spielbaren Version von <i>TRIVIA</i>	56
Abbildung 3.6 Screenshot aus <i>TRIVIA</i> mit Frage	56

## Glossar

### A

#### *Account*

Ein Account ist sozusagen der Zugang eines Spielers zu einem Onlinespiel. Er ist durch eine Kombination aus Benutzernamen und Passwort geschützt und sammelt alle Leistungen die der Spieler vollbringt. Ein Account kann je nach Spiel auch mehrere Spielcharaktere beinhalten.

#### *ActionScript*

ActionScript ist eine proprietäre Scriptsprache die von Macromedia für Flash entwickelt wurde um dynamische Animationen zu ermöglichen. Die Sprache basiert auf ECMA-Script und ähnelt damit ein wenig JavaScript. In der aktuellen Version 3.0 beinhaltet die Sprache einen Großteil der für objektorientierte Programmierung nötigen Elemente.

#### *AdSense*

Ist ein vom Suchmaschinenbetreiber Google entwickelter Dienst, der es einem Webseitenbetreiber ermöglicht, kontext-basierte Werbung auf seiner Seite zu schalten. Die Werbungen werden von Google zur Verfügung gestellt und der Webseitenbetreiber an den durch Klicks auf diese Werbe-Banner generierten Umsatz beteiligt.

#### *Alpha-Version*

Als Alpha-Version bezeichnet man die erste lauffähige Version eines Computerprogramms. Diese Version beinhaltet einen Großteil der Funktionalität ist jedoch meist noch nicht besonders stabil und enthält viele Fehler.

### B

#### *Beta-Version*

Die Beta-Version ist unfertige Version eines Computerprogramms und auch zumeist die erste die vom Hersteller zu Testzwecken veröffentlicht wird. Die Funktionalität ist meist bereits komplett verfügbar. Diese Version wird oft auch einem groß angelegten Test (beta test) unterzogen um Bugs zu finden.

### C

#### Client

In diesem Zusammenhang meint Client, eine Software die auf dem Rechner des Spielers installiert wird und die Präsentation eines Onlinespiels durchführt.

### E

#### Event handler

Ein event handler ist ein Element einer Scriptsprache. Es dient dazu, Ereignisse (wie z.B. einen Mausklick) abzufangen und eine entsprechende Aktion auszuführen.

**F**

## Flash

Flash ist eine Entwicklungsumgebung zur Erstellung von multimedialen Inhalten für Webseiten. Es zeichnet sich durch geringe Dateigrößen aus und ermöglicht mit Hilfe von ActionScript auch die Entwicklung von interaktiven Anwendungen.

**G**

## Grafikengine

Eine Grafikengine ist ein Framework das Grafikfunktionen für den Aufbau eines Spiels bereitstellt und dessen Funktionsumfang und Performance über die grafische Qualität des Spiels entscheidet. Meist werden fertige Grafikengines (z.B. Unreal-Engine) für ähnliche Spiele lizenziert und angepasst.

**I**

## IRC

IRC steht für Internet Relay Chat und bezeichnet ein textbasiertes Chatsystem. Es erlaubt eine beliebige Anzahl von Teilnehmern und es können Channels eingerichtet werden oder Einzelgespräche abgewickelt werden.

**M**

## Movieclip

Als Movieclips bezeichnet man die einzelnen Elemente einer Flash-Animation. Dabei kann man sich den Movieclip als eine Art Container vorstellen, der beliebigen Inhalt (auch andere Movieclips) kapselt. Dieser Container kann dann animiert werden, wobei sich die Änderungen (z.B. der Position) auf alle Elemente innerhalb des MovieClips auswirken. Jeder Movieclip verfügt über seine eigene Zeitleiste, die unabhängig von der Hauptzeitleiste abläuft.

## Multi-Hunter

Multi-Hunter sind Foren-Administratoren oder –Helfer deren Hauptaufgabe das Aufspüren von Spielern mit mehreren Accounts (Multis) ist. Das Erstellen mehrerer Accounts kann Vorteile im Spiel bringen, weswegen es bei manchen Spielen untersagt ist.

## MySQL

MySQL ist ein relationales Datenbanksystem auf Open-Source-Basis. Es bietet bis zu mittleren Projekten eine gleichwertige Funktionalität und Stabilität wie proprietäre RDBM-Systeme wie Microsoft SQL-Server oder Oracle. Als Bestandteil von LAMP (Linux, Apache, MySQL, PHP) – Systemen ist es heute auf den meisten Webservern vorhanden.

**P**

## Persistent World

Unter einer Persistent World versteht man eine ständig verfügbare Onlinewelt. Man kann sich somit jederzeit von überall in diese Welt einloggen und diese entwickelt sich auch weiter, wenn man selbst nicht daran teilnimmt. Die PW ist die Grundlage aller MMOGs.

**PHP**

PHP (PHP: Hypertext Preprocessor) ist eine freie, im Internetbereich weit verbreitete Scriptsprache zur Erstellung von dynamischen Webseiten

**PvP**

PvP steht für Player vs. Player und bezeichnet eine weit verbreitete Spielart in Onlinespielen, bei der sich Spieler gegenseitig bekämpfen.

**S****SWF**

SWF ist das Dateiformat für fertig kompilierte Flash-Filme. Diese Dateien können in eine Webseite eingebunden oder von einem Standalone-Player abgespielt werden.

**T****TCP/IP**

TCP/IP ist ein Internet-Transportprotokoll zum zuverlässigen, verbindungs-orientierten Austausch von Datenpaketen.

**Traffic**

Als Traffic wird jene Datenmenge bezeichnet, die durch Anfragen und Rückmeldungen eines Servers anfallen. Diese Datenmenge schlägt sich dann meist auch in den Kosten für den Betrieb eines Servers nieder.

**Type casting**

Unter type casting versteht man die Umwandlung einer Variable von einem Typ in einen anderen (z.B. von einer Gleitkommazahl auf einen Integer). Dies wird in typsicheren Sprachen benötigt, da jede Methode nur gewisse Variablentypen akzeptiert.

**U****UDP**

UDP ist ein Internetprotokoll zum verbindungslosen, unzuverlässigen Datenaustausch. Die Daten werden einfach geschickt, es wird keine Erfolgsmeldung abgewartet, wodurch sich ein geringer Overhead ergibt.

**X****XML**

XML steht für Extensible Markup Language ist ein Standard für maschinen- und menschenlesbare Dokumente in Form einer Baumstruktur.