

Technische und soziale Strukturen
virtueller Welten am Beispiel von
TubMud



Bericht 94-38

Irina Leyde

am Beispiel von TubMud

Bereits im Wintersemester 1989/1990 hatte ich davon gehört, daß am Fachbereich Informatik der TU-Berlin eine textbasierte multiuserfähige virtuelle Welt von Studentinnen und Studenten installiert wurde, die über Internet erreichbar war. Da ich befürchtete, daß eine nähere Beschäftigung mit dieser Welt zwangsläufig dazu führen würde, daß ich der Faszination der Kommunikation mit ihren ‚Bewohnern‘ erliege und mein Studium darunter leiden könnte, habe ich erst im Januar 1993 TubMud ‚betreten‘. Fast entsprechend meiner Befürchtung habe ich im vergangenen Jahr viel Zeit in dieser und anderen virtuellen Welten ‚verbracht‘. In dieser Zeit habe ich mit vielen Menschen aus unterschiedlichen Ländern ‚gesprochen‘ und interessante Beobachtungen sowohl über die technische Realisierung solcher Welten als auch über die Ausgestaltung der Gesellschaft in diesen Welten gemacht.

Da immer mehr Menschen immer besseren Zugang zu weltweiten Computernetzen bekommen und immer mehr Freizeit in virtuellen Welten verbringen, halte ich es für wichtig, sich mit der Ausgestaltung dieser Welten und daraus entstehenden Problemen auseinanderzusetzen. Dies möchte ich in meiner Studienarbeit tun.

In der Bundesrepublik ist bisher nur wenig über virtuelle Welten und die darin entstehenden sozialen Gefüge erschienen, so daß ich für meine Studienarbeit hauptsächlich auf englischsprachige Schriften zurückgegriffen habe.

Da ich selbst in einigen dieser virtuellen Welten, über die ich schreibe, involviert bin, mag mir vielleicht an mancher Stelle der nötige Abstand zum Forschungsobjekt fehlen. Andererseits kann ich aber auch nur dadurch, daß ich gewissermassen ‚Insider‘ bin, über manche Phänomene berichten, und hoffe, daß mein Betreuer als Unbeteiligter mir an dieser Stelle helfen wird.

Irina Leyde, November 1994
(ilian@cs.tu-berlin.de, Sharilla@TubMud)

Zur Veröffentlichung empfohlen von Prof. Dr. Bernd Lutterbeck

Nachsatz vom 10.04.1996

Die vorliegende Arbeit bezieht sich an vielen Stellen auf Beispiele und Beobachtungen aus TubMud, so zum Beispiel in den Kapiteln 2, 4 und 5, da dieses Mud lange mein „Heimatmud“ war. Trotzdem lassen sich meiner Meinung nach Parallelen zu anderen Muds finden und die Arbeit ist auch für LeserInnen aus diesen anderen Muds interessant.

Bei zukünftigen Arbeiten werde ich mich auf ein anderes Mud, Padermud bzw. Xyllomer (telnet 131.234.12.13 3000) beziehen und bedanke mich an dieser Stelle für die außerordentlich freundliche Aufnahme in Xyllomer.

Irina Leyde, April 1996
(ilian@cs.tu-berlin.de, ileyde@bb-data.de, Sharilla@Xyllomer, Sharilla@Tubmud and many other places)

Inhaltsverzeichnis

1	Einleitung	4
1.1	veni, vidi, vici...	4
1.2	Pflanzen, die im Verborgenen blühen...	5
2	Was ist eigentlich ein MUD?	6
2.1	Das Weltdorf des Internet	6
2.2	Das Eintauchen in eine fremde Welt...	7
2.3	Treten Sie näher!	8
2.4	Das Wandern...	9
2.5	Die Maus, die brüllte...	9
2.6	Wichtige Ergänzungen	10
2.7	Was macht man eigentlich in so einer virtuellen Welt?	11
2.7.1	Abenteuerorientierte MUDs	11
2.7.2	Nicht-abenteuerorientierte MUDs	12
3	Ursprung und Entwicklung der MUDs	14
4	Technische Konzepte	17
4.1	Ein Blick hinter die Kulissen	18
4.1.1	Objekte, Räume — alles ‚nur Dateien‘	19
4.1.2	Magie für Anfänger	20
4.2	Der Game Driver	21
4.3	Eine objektorientierte Welt	24
4.3.1	Ein Einblick in LPC	24
4.3.2	Objekte und Objekte und Objekte...	25
4.3.3	Kissen sind zum Werfen da!	27
4.3.4	LPC für Fortgeschrittene	28
4.4	An’s Licht, zur Sonne...	29
5	Gesellschaft en miniature	33
5.1	Aufbau des sozialen Mikrokosmos TubMud	34
5.1.1	Die Gruppe der Wizards	34
5.1.2	Die Gruppe der Peers	37
5.2	Was machen Leute im MUD?	38
5.2.1	Kommunikation ohne Stimme	39

5.2.2	Ich möchte ein Eisbär sein	40
5.2.3	Der Stoff aus dem die Träume sind	42
5.2.4	The bright side of life?	43
5.3	Manchmal sind Anarchie und Diktatur Schwestern	45
5.4	Ein Ausblick	49
6	Schlußbetrachtung	51
	Literaturverzeichnis	54
A	Die Levelstruktur	58
B	Die Wizards' rules	59
C	Die Players' rules	62
D	Das Agreement	64
E	Die Schließungsmeldung von TappMud	68
F	Ein Programmbeispiel	69

Abbildungsverzeichnis

4.1	Bestandteile des Game Drivers	22
4.2	Ein Beispiel für Mehrfachvererbung in LPC.	25
4.3	Zusammenwirken von Objekten	31
4.4	Vorgänge beim Einloggen	32
5.1	Aufbau der TubMud-Gesellschaft	35

Kapitel 1

Einleitung

1.1 veni, vidi, vici. . .

Der Einzug der Computer in den privaten Bereich der Menschen ist schon lange vollzogen. In vielen Haushalten ist er schon fast nicht mehr wegzudenken, sei es als Ersatz für Schreibmaschine und Haushaltsbuch oder als Freizeitbeschäftigung. Vielfach hat er den Büchern, Spielen und sogar dem Fernseher den Rang abgelaufen und häufig sitzen die Menschen allein vor dem Monitor und versuchen, kleine grünhaarige Zwerge zum rettenden Ausgang eines Labyrinths zu dirigieren oder ein imaginäres Flugzeug zu fliegen.

Seit die Computer Anfang der 80er Jahre weitere Verbreitung in den privaten Haushalten gefunden haben, hat es sich bei den Freizeitspielen meist um Einzelspiele gehandelt. Erst in den letzten Jahren kamen Spiele auf, für die eine Verbindung der Computer vorgesehen ist und bei denen je nachdem miteinander oder gegeneinander angetreten wird. Allen diesen Spielen gemeinsam ist aber, dass innerhalb der Spiele keine Kommunikation stattfindet, sondern höchstens außerhalb, wenn die Spieler im gleichen Raum sitzen.

Mit den zunehmenden technischen Möglichkeiten gerade im Graphikbereich der neuen Computergenerationen und der Entwicklung sogenannter Datenhandschuhe und Sichtbrillen, traten **virtuelle Welten** auf spektakuläre Weise in den Mittelpunkt der Medien und Computermessen¹. Bei diesen handelt es sich meist um sehr aufwendige Systeme, sowohl in technischer als auch in finanzieller Sicht. Der Spielende sitzt nicht mehr vor einer Spielekonsole, sondern kann über zwei Monitore vor den Augen, die in einer Spezialbrille integriert sind, programmierte Räume nahezu dreidimensional sehen und mit Hilfe des Datenhandschuhs Gegenstände im virtuellen Raum berühren und manipulieren. In diesem Datenhandschuh werden mittels eines Sensorsystems selbst kleinste Bewegungen der Hand registriert und als Eingaben für das Programm ausgewertet. Befinden sich mehrere an das System der virtuellen Welt angeschlossene Personen im gleichen virtuellen Raum, können sie miteinander interagieren. Bei geeigneter Vernetzung müssen sich die Spielenden nicht einmal im gleichen realen Raum befinden und können sich in der realen Welt sogar völlig fremd sein.

¹Ich verweise auf die Artikel in: Der Spiegel [6], [8],[9], [10], c.t [12], PowerPlay [13] Die Zeit [11]

1.2 Pflanzen, die im Verborgenen blühen...

Obwohl in aller Munde, sind die Virtual Reality Systeme (auch Cyberspace genannt) dank ihres großen technischen Aufwandes und der sehr hohen Preise bisher noch nicht sehr verbreitet. Die Technik ist kompliziert, störungsanfällig, teuer — kurzgesagt, etwas für Leute mit der nötigen Hingabe und gut gefülltem Geldbeutel.

Viel unspektakulärer und daher auch unbemerkter verlief die Entwicklung textbasierter multiuserfähiger virtueller Welten, sogenannter MUDs, von denen bereits 1977 frühe Versionen im universitären Bereich existierten. Bei den textbasierten MUDs findet die Interaktion zwischen Spielendem und Programm ausschließlich über Text statt. Informationen werden als Text auf dem Bildschirm ausgegeben und Eingaben erfolgen über die Tastatur. Es gibt zwar mittlerweile auch Versuche mit grafikorientierten MUDs, allerdings sind diese im Vergleich mit den textbasierten deutlich in der Minderzahl. Im Kapitel 4 werde ich noch näher auf grafikorientierte MUDs eingehen.

Ich benutze den Begriff MUD, der für **M**ulti-**U**ser **D**ungeon steht, als Oberbegriff für die mittlerweile vielen unterschiedlichen Arten von virtuellen Welten. Sie lassen sich in MUSEs, MUSHs, MOOs, MUDs und andere Kategorien einteilen, auf die ich später noch eingehen werde.

Heute sind ungefähr 300 dieser MUDs über Internet weltweit erreichbar und eine große Zahl von Menschen verbringt ihre Zeit in ihnen. Manche dieser MUDs sind nicht als Freizeitbeschäftigung konzipiert, sondern dienen wissenschaftlichen Zwecken. In diesem Fall ist der Zugang meist begrenzt und Studierende oder wissenschaftlich Arbeitende des gleichen Fachgebietes oder zumindest ähnlicher Forschungsgebiete vorbehalten.

Kapitel 2

Was ist eigentlich ein MUD?

2.1 Das Weltdorf des Internet

Die meisten MUDs sind, wie schon erwähnt, über **Internet**¹ erreichbar. Das Internet, eigentlich ‚inter networking‘, geht auf das ARPAnet der frühen siebziger Jahre zurück und wurde von der DARPA (Defense Advanced Research Projects Agency) [23] initiiert. Es hat seinen Ursprung somit sowohl im universitären als auch im militärischen Bereich. Das Internet ist der weltweit größte Verbund von Computern, ein internationales Datennetz, das heute rund 25 Millionen Menschen zur Kommunikation benutzen².

Mit Hilfe des Übertragungsprotokolls telnet oder eines anderen geeigneten Programms, z. B. einem sogenannten MUD-Frontend, der Internetadresse und der Portnummer kann man sich in einem MUD einloggen. Auf diese Art und Weise kann man mit dem Befehl telnet morgen.cs.tu-berlin.de 7680 TubMud an der TU-Berlin erreichen.

Die meisten MUDs, so auch TubMud, sind englischsprachig. Das ist für Spieler mit geringen Englischkenntnissen schwierig, aber auf diese Art und Weise können Menschen aus unterschiedlichen Ländern ohne zusätzliche Absprachen miteinander über das Medium Internet und somit auch in MUDs kommunizieren. Wenn man sich das erste Mal in einem MUD einloggt, erschafft man eine virtuelle Person, einen **Character**. Man kann für gewöhnlich Namen und Geschlecht des Characters wählen, in manchen MUDs auch eine Beschreibung. Das Geschlecht ist frei wählbar, unabhängig vom realen Geschlecht des Spielenden. Es stehen weiblich, männlich und zuweilen auch neutral oder plural zur Verfügung. Ein neutraler Character kann zum Beispiel ein Tier oder eine Pflanze sein, als Plural Character ist ein Bienenschwarm oder Tim-und-Struppi denkbar.

MUDs sind meist Abbilder des tatsächlichen physikalischen Raumes. Zu den einzelnen Räumen der virtuellen Welt existieren Raumbeschreibungen, mit deren Hilfe man sich die Räume, Plätze oder Häuser möglichst einfach und bildlich vorstellen kann. Im Lauf der Zeit tut die Phantasie ein übriges und man beginnt, die Räume zu ‚sehen‘.

¹Siehe hierzu Kehoe: [20] und: Gaffin/Heitkoetter [30]

²Artikel in: Der Spiegel [7]

2.2 Das Eintauchen in eine fremde Welt...

Wenn man sich in TubMud einloggt, dann bekommt man als erstes die folgende Raumbeschreibung:

You feel full of awe as you meet Hermes.

The portal of the village church is wide open. Dozens of tiny lights illuminate a dome at the far end of the church. The tower of the church lies east of the entrance. The place in front of the church is bordered by a pit to the west and the main road, crossing the meadow to the south.

A note is pinned to the signpost beside the portal.

There are two obvious exits: south and north.

Jerdian the witch (nice)

Hermes the messenger of the gods.

Auf die Raumbeschreibung folgt die Aufzählung der ‚offensichtlichen‘ Ausgänge. Dies ist zum einen gut dafür, sich den Raum vorzustellen, ist aber gleichzeitig auch eine Andeutung, daß es ja noch Ausgänge geben könnte, die nicht so ganz offensichtlich sind und die man erst suchen müßte.

Eine Raumbeschreibung erhält man, wenn man einen Raum neu betritt, z.B. beim Einloggen, oder mit dem Befehl `look`. Betritt ein Spieler, z.B. Marvin, einen Raum, in dem sich andere Spieler befinden, so bekommen diese eine Mitteilung, die etwa so aussieht:

Marvin arrives.

Nach der Raumbeschreibung und der Liste der offensichtlichen Ausgänge folgt eine Aufzählung der sich im selben Raum befindlichen anderen Spieler. In unserem Fall befindet sich außer dem eigenen Character noch eine Spielerin namens Jerdian, die den Titel `witch`³ führt, auf dem Platz vor der Kirche und außerdem ein Spieler namens Hermes. Der Ausdruck `nice` in Klammern hinter Jerdians Titel gibt an, daß sie eine anderen Wesen eher freundlich gesonnene Hexe ist. Es handelt sich hierbei um das sogenannte **Alignment**, das sich je nachdem, wie man sich verhält, zwischen `saintly` und `demonic` bewegen kann. Hermes, der sich ebenfalls im gleichen virtuellen Raum aufhält, ist kein richtiger Mitspieler. Bei ihm handelt es sich um eine programmierte Figur, die bestimmte für das Spiel wichtige Aufgaben wahrnehmen kann. Programmierte Spieler, deren Verhalten also nicht direkt von einem am Terminal sitzenden Menschen beeinflußt werden, heißen NPC, **Non-Player-Character**. Tatsächliche Mitspieler und NPCs lassen sich nicht immer leicht unterscheiden, da NPCs oft so programmiert sind, daß sie mit tatsächlichen Spielern umfangreich interagieren können.

³Mit etwas Erfahrung kann man darauf schließen, daß Jerdian schon eine Weile spielt. Ein Verzeichnis der zu den Leveln gehörigen Titel liegt als Anhang bei.

Hilfstexte zu den einem zur Verfügung stehenden Befehlen kann man mit help abrufen. Man erhält den folgenden Hilfstext:

```
> help
COMMUNICATIONS:
say <msg>, shout <msg>, tell <player> <msg>, converse, emote <text>
INFO:
score, hp, help, who, i (inventory), email
OTHER:
password      typo <msg>brief      get <obj> [from <obj>]
save          bug <msg>wimpy      give <obj> to <obj>
quit          typo <msg>          put <obj> into <obj>
ATTACKS:
kill <name>, stop (hunting), <spell> <monster>
OTHER TOPICS:
alignment  experience  levels      source      start      wear
drop       hitpoints  put         spell       stats      wield
dying      killing    remove     spellpoints syntax
NEW TOPICS:
quests party skills bandits mudlink webster explore soul feelings adverbs
```

2.3 Treten Sie näher!

Genauso, wie man sich einen Raum mit Hilfe des Befehls look ansehen kann, kann man sich auch Dinge und andere Spieler im gleichen Raum näher ansehen. Mit dem Befehl look at sehen wir uns die Kirche näher an:

```
> look at church
You'll be able to see more from inside.
```

Das brachte keine neuen Erkenntnisse, versuchen wir's weiter:

```
> look at tower
There is a towerclock high above you.
```

Am Kirchturm hängt eine Uhr und wir können versuchen, uns auch diese näher anzusehen:

```
> look at clock
In the bright sunlight you read: twenty-three minutes to three.
A small clock below is labeled 'Uptime': 8 days 2 hours 35 minutes 52 seconds
```

Auf diese Weise kann man nun den jeweiligen virtuellen Raum, in dem man sich aufhält, und darin befindliche Gegenstände oder Mitspieler näher betrachten. Man bekommt mehr und mehr ein Bild des virtuellen Raums.

2.4 Das Wandern...

Um sich von einem Raum in den nächsten, angrenzenden, zu bewegen, gibt man einfach die Himmelsrichtung an, in die man gehen will. In der Liste der offensichtlichen Ausgänge sind die möglichen Bewegungsrichtungen angegeben. Manchmal kann man auch nach oben oder unten gehen oder einen Raum oder z.B. eine Kutsche mit `enter` betreten. Wie schon angedeutet, sind nicht immer alle Ausgänge eines Raumes sofort erkennbar. Eine Geheimtür findet man zum Beispiel erst nach näherer Untersuchung des Raumes, sie würde auch nicht in der Liste der offensichtlichen Ausgänge erscheinen.

2.5 Die Maus, die brüllte...

Laut Curtis⁴ verbringen die meisten Spieler die meiste Zeit in MUDs damit, miteinander zu kommunizieren. Ich kann dies nur bestätigen. Und tatsächlich steht einem für die Kommunikation eine Vielzahl von Möglichkeiten zur Verfügung.

Für die sprachliche Kommunikation kann man insbesondere die Befehle `say`, `whisper`, `tell` und `shout` benutzen.

Gibt Marvin den Befehl

```
> say Hallo!
```

ein, dann bekommen alle anderen Spieler, die sich im selben Raum befinden, die Zeile

```
Marvin says: Hallo!
```

auf den Bildschirm.

Marvin selbst bekommt eine Bestätigung des Geäußerten in Form einer Wiederholung. Benutzt Marvin anstatt `say` den Befehl

```
> whisper Jerdian Wo ist denn die Post?
```

dann wird die Frage nur Jerdian mitgeteilt. Alle anderen im Raum befindlichen Spieler merken aber natürlich, daß geflüstert wird:

```
Marvin whispers something to Jerdian.
```

Unauffälliger und auch geeignet, wenn sich die Spieler, die miteinander reden wollen, nicht im selben Raum befinden, ist der Befehl `tell`. Bei `tell` bekommt nur die Zielperson die Information und niemand sonst merkt etwas von dieser Kommunikation. Ganz im Gegenteil dazu hört ein

```
> shout Auf Wiedersehen!
```

jeder in TubMud eingeloggte Spieler.

⁴'The majority of players spend the majority of their active time on MUDs in conversation with other players.' [4], Seite 11/21

Neben den zur sprachlichen Kommunikation vorgesehen Befehlen gibt es noch eine Menge anderer, die zum Ausdrücken von Gefühlen geeignet sind. Es stehen den Spielenden ungefähr 100 Verben zur Verfügung, die mit fast 200 Adverbien kombiniert werden können. Ein `laugh` von Jerdian führt für sie zu der Mitteilung:

You laugh.

und für alle anderen im Raum befindlichen Mitspieler:

Jerdian laughs.

Kombiniert mit dem Adverb `friendly` und dem Mitspieler Marvin ergibt sich für Jerdian:

```
> laugh Marvin friendly
You laugh friendly at Marvin.
```

Marvin bemerkt:

Jerdian friendly laughs at you.

und die anderen Mitspieler im Raum sehen:

Jerdian friendly laughs at Marvin.

Durch die Vielzahl von Verben und Adverbien werden die Unterhaltungen plastischer. Die Spieler sind nicht nur auf Worte angewiesen, sondern haben auch Gestik und Mimik zur Verfügung, wenn auch im Vergleich zur realen Welt in eingeschränktem, schematisiertem Maße.

2.6 Wichtige Ergänzungen

Von den vielen der Information des Spielers dienenden Befehlen möchte ich hier nur das `who` erwähnen, mit dem man (analog zu UNIX) erfahren kann, wer zur Zeit am Spiel teilnimmt. Man erhält als Antwort:

```
-----
/                                     / \
|               There are 9 players on TubMud.               | |
\-----\-----\-----\-----\
|
| Demos the archmage (old and weary)                          |
| Devotio is always willing to die for Sharilla (testing)     |
| Hal the 9000th, Domainlord of Snowdonia (busily LaTeXing for Sharilla) |
| Jerdian the witch (nice)                                     |
| Logon the titleless (logging on)                            |
| Marvin the depressive robot [13] (evil)                       |
| Fluffy Sharilla the incredible dangerous Heidschnucke (sheeping around) |
| Stone statue of ghost of Swampnose (rock watch material)   | -
| Webster the English Professor (explaining words you tell him) | |
\-----\-----\-----\-----\
```

Passend zu der mittelalterlichen Sagenwelt, die TubMud darstellt, hat die Liste der Spielteilnehmer die Form einer Schriftrolle. Obwohl es sich um eine einfache Grafik aus ASCII-Zeichen handelt, stellt sie ein Durchbrechen der rein textbasierten Technik dar; den Versuch, den Bildschirm in eine Schriftrolle zu verwandeln und den Spielenden noch ein wenig mehr aus der realen Welt in die des MUDs zu ziehen.

2.7 Was macht man eigentlich in so einer virtuellen Welt?

Die Möglichkeiten der Beschäftigung in einem MUD hängen zum größten Teil davon ab, um was für eine Art virtueller Welt es sich handelt. Die zur Zeit ungefähr 300 weltweit existierenden MUDs unterscheiden sich alle voneinander. Auf die genaueren Unterschiede gehe ich im folgenden noch ein, ich möchte hier, analog zu Bruckman⁵ erst einmal zwischen abenteuerorientierten und nicht-abenteuerorientierten MUDs unterscheiden. Diese Unterteilung ergibt sich aus dem Ursprung der MUDs, den ich in Kapitel 3 noch näher erläutern werde. Bemerkenswert ist, daß die Idee der MUDs aus dem Bereich der Rollenspiele kommt, bei denen es darum geht, mit einer Gruppe von Spielern eine gestellte Aufgabe, zum Beispiel Lösen eines Rätsels oder Beschaffen eines Gegenstandes, zu erfüllen. Dabei handeln die Spielenden mittels imaginärer Personen in einer imaginären Welt und agieren, indem sie dem Spielleiter mitteilen, was sie tun. Der Spielleiter kennt den Hintergrund des Puzzles und handelt auch für alle NPCs. Je nach Ausgestaltung der imaginären Welt stehen Spielern und NPCs die Möglichkeiten von Magie (z. B. in einer Fantasy-Welt), Raumfahrt (in einer Science-Fiction-Welt) oder sogar Zeitreisen zur Verfügung. Das Rollenspiel basiert auf der Kommunikation und Fantasie der Spielenden. Als Beispiel mag hier das schon sehr lange existierende Rollenspielsystem D&D — Dungeons & Dragons⁶ dienen, das für viele frühe Computer-Spiele die Vorlage war und in dem sich die Spielenden ebenfalls in einer mittelalterlichen Sagenwelt voller Ritter, Zwerge, Elfen, Drachen und Magier bewegen.

2.7.1 Abenteuerorientierte MUDs

Dem klassischen Rollenspiel sehr nahe kommen die abenteuerorientierten MUDs. Die Rolle des Spielleiters wird hier von MUD-Programmen (unter anderem vom sogenannten Gamedriver, auf den ich noch näher eingehen werde) übernommen, außerdem müssen die Spielenden, wie schon erwähnt, nicht beieinandersitzen.

Als abenteuerorientierte MUDs sind hier die MUDs der LP-Familie, zu denen auch TubMud gehört, sowie die Diku-, Unter- und AberMUDs zu erwähnen⁷. Die ungewöhnlichen Namen der MUD-Typen gehen teilweise auf die Namen der Programmierer oder den Ort, an dem das jeweilige MUD entwickelt wurde, zurück.

⁵,While every MUD is different, there are two basic types: those which are like adventure games, and those which are not.' [1]

⁶siehe hierzu: Spielerhandbuch [16]

⁷siehe hierzu: Frequently Asked Questions [14] und Bruckman [1]

Das Ziel des Spiels in abenteuerorientierten MUDs ist es, Rätsel zu lösen, Monster zu überwinden, Schätze zu finden und auf diese Art und Weise Erfahrung in Form von Erfahrungspunkten zu gewinnen.⁸ Trotzdem stehen die Spielenden natürlich auch in diesen MUDs häufig im Gespräch beieinander, und Kommunikation spielt eine bedeutende Rolle für die meisten Teilnehmer.

Ein Spieler, der Erfahrungspunkte sammelt, kann in höhere Erfahrungsstufen aufsteigen und dadurch seine Handlungsmöglichkeiten im Spiel erweitern, indem er z. B. neue Zaubersprüche lernt. In den meisten abenteuerorientierten MUDs kann der Spieler nach dem Sammeln einer bestimmten Menge von Erfahrungspunkten und Lösen von Rätseln aufsteigen und Wizard werden. Zu diesem Zeitpunkt hat der Spielende oft mehr als einhundert Stunden ‚MUDdend‘ verbracht und den größten Teil der virtuellen Räume besucht.

Wizards sind Spieler, die quasi das Spiel gewonnen haben. Sie haben Zugriff auf die Datenbasis des MUDs⁹, können sie erweitern und zu der virtuellen Welt neue Räume, zu lösende Rätsel, NPCs oder andere Objekte beisteuern. Obwohl Wizards auf diese Weise sehr mächtig sind, unterliegen sie doch strengen Verhaltensregeln (Wizards' rules) um zu verhindern, daß sie ihre Fähigkeiten mißbrauchen und anderen Mitspielenden das Spiel verderben. Die Wizard's Rules, die im Anhang zu finden sind, sowie eine Vielzahl anderer geschriebener und ungeschriebener Regeln, wer sie einhalten muß und wer sich darüber hinwegsetzen kann, spielen eine extrem wichtige Rolle für das soziale Gefüge einer virtuellen Welt, das in Kapitel 5 eingehend untersucht wird.

Die Leitung des MUDs obliegt dem Gott, der absolute Macht hat und nach Gutdünken handeln kann. Innerhalb der Wizards gibt es häufig noch Hierarchieabstufungen entsprechend der Einflußnahme auf technische und politische Entscheidungen betreffend die Entwicklung des MUDs. Wizards höheren Ranges werden von der Leitung des MUDs auf ihren Rang benannt. Sozialer Status und Rang beeinflussen sich in starkem Maße gegenseitig. Ein Wizard, der interessante Objekte programmiert, sich eventuell für das soziale Gefüge im MUD, die ‚MUD-Gesellschaft‘, einsetzt und dadurch ein gewisses Ansehen genießt, wird innerhalb kurzer Zeit in einen höheren Rang erhoben werden. Gleichzeitig führt aber auch ein höherer Rang, obwohl einfach durch die Leitung des MUDs benannt, im allgemeinen zu einer Steigerung des Ansehens.

Verfehlungen gegen den Ehrenkodex, den die Wizards' Rules darstellen, werden analog zur Rangordnung geahndet. Verstößt ein Wizard gegen den Ehrenkodex, so wird er von Wizards höheren Ranges mit Sanktionen belegt. Verstöße von Wizards mit hohem Rang werden nur noch innerhalb der Ranggruppe geregelt und nur in den seltensten Fällen offen diskutiert. Das soziale Gefüge in den meisten abenteuerorientierten MUDs ist tatsächlich streng hierarchisch¹⁰ geordnet.

2.7.2 Nicht-abenteuerorientierte MUDs

Die nicht abenteuerorientierten MUDs, zu denen die Tiny- und Teeny-MUDs¹¹ gehören, haben eine andere Philosophie. Bei ihnen stehen die sozialen Kontakte, die die Spielenden

⁸Zu diesem Punkt: Frequently Asked Questions [14], Teil 1, Punkt 2

⁹Hierzu auch: Frequently Asked Questions [14], Teil 1, Punkt 34

¹⁰Nachzulesen bei Bruckman [1], Kapitel 2, Seite 6

¹¹siehe Frequently Asked Questions [14]: Teil 1, Punkt 3 und Bruckman [1], Seite 11

im MUD knüpfen, im Vordergrund. Es gibt zwar auch zu lösende Puzzle und zu besuchende Orte, aber das kämpferische Rollenspielelement ist so gut wie ausgeblendet. Man kann das Spiel nicht gewinnen, weil es gewissermaßen kein Ende hat.

In den meisten nicht abenteuerorientierten MUDs kann jeder Mitspieler auf die Datenbasis des MUDs zugreifen und neue Objekte und Räume kreieren. Im Gegensatz zu den abenteuerorientierten MUDs, bei denen der Status eines Spielers aus der Kenntnis der Geographie der Welt, der Rätsel und der Monster herrührt, erwirbt sich ein Spielender in dieser Form der MUDs Ansehen in Gesprächen und durch die Entwicklung interessanter Gegenstände oder Räume.

Ein grundlegender Unterschied zu den abenteuerorientierten MUDs ist laut Bruckman [1] (Kapitel 2.3) der liberale und egalitäre Hintergrund, vor dem die Spielenden agieren. Zwar gibt es aus technischen Gründen eine MUD-Leitung, aber vom Grundsatz her sollen doch alle Spielenden gleich sein, Mitspieler desselben Spiels.

Das miteinander reden, diskutieren, zusammen Spaß und ‚Sich treffen‘ haben steht im Mittelpunkt dieser MUDs. In abenteuerorientierten MUDs wird zwar auch sehr viel geredet, diskutiert und gescherzt, aber es gibt dort auch immer wieder Mitspieler, die an diesen lockeren, spontanen Treffen nicht teilnehmen, sondern nur durch die virtuelle Welt ziehen und Monster ‚schlachten‘ oder Rätsel lösen.

Eines der schönsten Beispiele für nicht abenteuerorientierte MUDs ist das von Pavel Curtis betreute und beschriebene ¹² **LambdaMoo**.

Manche der nicht abenteuerorientierten MUDs sind in literarischen Welten angesiedelt. So gibt es zum Beispiel **TrekMUSE** ¹³, das an der Fernsehserie **Star Trek — The Next Generation** orientiert ist und in dem man sich auf Raumschiffen und Raumstationen befindet. Man kann sich aber auch vorstellen, daß eine dieser Welten wie die der ‚Alice im Wunderland‘ aussieht, in der es ja nun genug wundersame Dinge gibt, die man zum Leben erwecken kann. In themenorientierten MUDs gibt es, vermutlich weil die literarischen Vorlagen so aufgebaut sind, in den meisten Fällen ein hierarchisches System.

Die Unterschiede zwischen den abenteuerorientierten und den mehr MUD-gesellschaftlich orientierten MUDs sind im Lauf der Entwicklungsgeschichte dieser Form der virtuellen Welten entstanden.

¹²siehe:[4] LambdaMOO. Es ist erreichbar unter lambda.parc.xerox.com 8888 bzw. 3.2.116.36 8888

¹³TrekMUSE ist erreichbar unter excalibur.mit.edu 1701 bzw. 18.80.0.247 1701

„... and don't despair if you die, you haven't lived until you've died in this MUD.“

DEMOS, God in TubMud

Kapitel 3

Ursprung und Entwicklung der MUDs

Wie schon erwähnt, liegt der Ursprung der MUDs sowie der übrigen Computer Adventure Games in den im Lauf der siebziger Jahre verstärkt in Mode gekommenen Rollenspielen, von denen ihrerseits viele auf Tolkiens Fantasyroman ‚Der Herr der Ringe‘ basieren. So erkennt man denn auch in Home-Computer-Spielen wie *Wizardry* oder *Bard's Tale* deutlich die Verwandtschaft zum frühen Rollenspielsystem *D&D — Dungeons & Dragons* von Gary Gygax und Dave Arneson¹. Alle diese Home-Computer-Spiele sind immer nur für jeweils einen Spielenden vorgesehen.

Bereits vor der kommerziellen Entwicklung von Adventure-Games für Homecomputer wurde im universitären Bereich in mehreren Ländern mit dieser Art von Spielen experimentiert. In den frühen siebziger Jahren entdeckte Donald Woods im Palo Alto Research Centre der Xerox Corporation ein Computerspiel, daß sich von allen bisherigen unterschied. Es galt nicht, graphisch dargestellte Raumschiffe oder ähnliches zu treffen, sondern ein imaginäres Labyrinth zu erkunden und Schätze zu finden. Das Spiel bestand nur aus den Beschreibungen der Räumlichkeiten und Gegestände. Woods nahm Kontakt zu dem Entwickler des Spiels, Will Cowther, auf und entwickelte es weiter. Woods programmierte *Advent* als eine Sagenwelt nach dem Vorbild Tolkiens, in der der Spieler Monster und Hindernisse überwinden und durch die virtuelle Welt wandern konnte. Im Gegensatz zu späteren Entwicklungen hatte *Advent* kein erklärtes Ziel neben dem Erkunden der Welt. Da *Advent* sehr schnell eine Reihe von Anhängern fand, wurden bereits kurze Zeit später ähnliche Spiele programmiert, die aber auch zunächst nur für einen einzelnen Spieler konzipiert waren.²

Eines der sehr frühen Spiele, das schnell auch für erste Mikrocomputer zur Verfügung stand, war das von Marc Blank, Tim Anderson, Bruce Daniels und David Lebling entwickelte *Zork* [17]. Es lief zunächst auf einer DEC PDP-10, aber bald wurden Versionen für Apple II, PDP-11 und TRS-80 entwickelt. Um das Spiel möglichst unabhängig von der jeweiligen Rechnerarchitektur eines einzelnen Mikrocomputers zu machen, wurden für unterschiedliche Rechnertypen virtuelle Maschinen entwickelt, die den *Zork*-Code interpretieren konnten. *Zork* war für einen Spielenden konzipiert, stellte aber bereits umfang-

¹siehe hierzu Vorwort des AD&D-Spielerhandbuchs [16]

²siehe hierzu: Reid [3], Seite 12,13

reiche Möglichkeiten, in der virtuellen Welt aktiv zu werden, zur Verfügung und kann als einer der Vorläufer der heutigen MUDs angesehen werden.

Laut Bruckman³ liefen die ersten MUDs auf Mehrplatz-Systemen und erlaubten mehrere Mitspielende. Es waren wohl hauptsächlich Studierende, die die Möglichkeiten der an den Universitäten vorhandenen Großrechner nutzen wollten, um Rollenspiele auf Computersysteme zu übertragen. Einige dieser Studenten und Studentinnen jobbten in kommerziellen Spielesoftwarefirmen, wo ihnen ihre Erfahrungen mit MUDs halfen, Home-Computer-Spiele zu entwickeln, die dann wiederum nur von einem Spieler allein gespielt werden konnten.

Nach dem Ausbau von Einzelbenutzer- zu Mehrbenutzerspielen war die nächste Hürde der Anschluß an Computernetze. Eines der ersten netzfähigen Spiele dürfte das 1977 von Jim Guyton für ARPAnet adaptierte *mazewar* gewesen sein. In diesem Spiel war noch keine Kommunikation unter den Mitspielern möglich. Man konnte sich vielmehr nur in einem Labyrinth gegenseitig jagen und abschießen. Das kurze Zeit später erschienene *Wizard* wies erstmalig Konzepte von Spielerinteraktion auf, die über Schußwechsel hinausgingen.⁴ Auf dem D&D-Rollenspiel-System basieren unter anderem die 1978-1979 erschienenen Spiele *Scepter of gods* von Alan Klietz und *MUD1* von Richard Bartle und Roy Trubshaw. *Scepter of gods* war sozusagen als Mehrspielervariante zu *Advent* konzipiert. Es hatte Möglichkeiten zur Kommunikation und die Spielenden konnten sowohl miteinander, als auch gegeneinander spielen. Der Name MUD für Multi-User-Dungeon wird erstmals von Trubshaw verwendet, der damit an das frühe Single-User-Spiel *Dungen* erinnern will. Wie man bei Bartle⁵ nachlesen kann, lief *MUD1* auf einem DECsystem-10 Großrechner der Essex University und konnte nur von Studenten und Studentinnen vor Ort gespielt werden. Nach einiger Zeit war es dann auch für außeruniversitäre Spielende möglich, sich per Modem einzuloggen. Bartle selbst bezeichnet die heute existierenden MUDs als Abkömmlinge des von ihm mitentwickelten *MUD1*⁶, das er als erstes ‚ordentlich‘ arbeitendes MUD ansieht.⁷ Dies kann ich anhand der bisher skizzierten Entwicklungsgeschichte der MUDs und der von mir benutzten Literatur nicht bestätigen.

Für wahrscheinlicher halte ich, daß an Universitäten unterschiedlicher Länder zu ungefähr gleicher Zeit Versuche unternommen wurden, single-user-Spiele multi-user- und netzfähig zu machen und sie weiterzuentwickeln. Dabei haben sich die unterschiedlichen Konzepte mit Sicherheit gegenseitig beeinflußt (schon allein, weil MUD-Autoren häufig gleichzeitig Spielende in anderen MUDs waren), wodurch sich naturgemäß gewisse Ähnlichkeiten ergeben. Allerdings haben sich im Verlauf der Entwicklung auch technische Unterschiede herausgebildet, die eine Rückführung aller heute existierenden MUDs auf ein bestimmtes einzelnes MUD nicht erlauben⁸, bzw. anmaßend erscheinen lassen. Jemand, der Spielen-

³siehe hierzu Bruckman: [1], Seite 6, Fußnote 21

⁴siehe Reid: [3], Seite 14

⁵[15], Teil 1.6 und 1.7

⁶‚Present day MUAs are all descendents of a single game known as MUD (Multi-User Dungeon; to avoid confusion with the generic term, the game will be referred to as MUD1 for the remainder of this report).‘ [15], Seite 6

⁷‚Although there were early attempts to turn single-player adventures [...] into multi-player adventures [...] these came to nothing or petered out. MUD1 was the first proper, workable multi-user adventure game.‘ [15], Seite 6

⁸hierzu: ‚LPMUD plays as a good MUD1 clone until wizlevel is reached.‘ [15], Seite 77 und folgende

der in einem MUD ist und dann selbst eines entwirft und implementiert, hat nach meiner Auffassung damit nicht automatisch einen Klon des oder der MUDs geschaffen, in denen er gespielt hat.

Von den unterschiedlichen, heute existierenden Familien von MUDs betrachte ich im folgenden die LPMuds, zu denen auch das von mir gewählte Beispiel TubMud gehört, genauer. Die LPMuds gehen auf die von einer Gruppe um Alan Cox herum entwickelten AberMuds zurück. Wie Alan Cox in seinem Artikel ‘A history of Muds’ [19] beschreibt, spielte er anfänglich MUD1 und schrieb dann zusammen mit anderen Studenten mehrere Versionen eines eigenen MUDs, das AberMud genannt wurde. Obwohl Bartle die schlechte Implementation und Ausgestaltung von AberMud kritisiert⁹, so spricht doch eine gewisse Verbreitung und Popularität, die auch er zugibt, dafür, daß die AberMuds eine qualitative Weiterentwicklung darstellen.

Lars Pensjö, selbst Spieler von AberMud und TinyMud, wollte die beiden Systeme kombinieren und entwickelte sein LPMud¹⁰. Die LPMuds weisen einige entscheidende Verbesserungen und Neuerungen gegenüber den vorherigen Entwicklungen auf, die ich in Kapitel 4 noch näher betrachten werde. Pensjö entwickelte für sein MUD die objektorientierte Sprache LPC (Lars Pensjö C), die zur Erweiterung der Datenbasis benutzt wird. Das von Pensjö ins Leben gerufene und betreute MUD Genesis¹¹ läuft noch heute und zuweilen kann man ihn dort antreffen, obwohl er sich aus der aktiven Arbeit an MUDs zurückgezogen hat.

Auch heute ist die Entwicklung der MUDs noch nicht abgeschlossen. Im Wintersemester 1993/94 startete an der TU Berlin das autonome Seminar¹² a_S D.O.M.E mit der Zielsetzung, einen Game Driver, das Herzstück eines jeden MUDs, als verteiltes System zu entwickeln und implementieren.¹³ Außerdem gibt es mit MediaMOO von Amy Bruckman¹⁴ und Astro-VR von Pavel Curtis und David A. Nichols¹⁵ ernstzunehmende Versuche, MUDs für wissenschaftliche Zwecke nutzbar zu machen und gemäß dem heutigen Stand der Videokonferenztechnik weiterzuentwickeln.

⁹siehe hierzu: [15] Seite 75/76

¹⁰, In the beginning, I played a lot of Abermud and some Tinymud, and wanted to do something better, combining the two systems. I made the first version of LPmud as some kind of argument, to show that my ideas were possible.’ [26]

¹¹Genesis ist erreichbar unter milou.cd.chalmers.se.

¹²Ein autonomes Seminar ist eine universitäre Lehrform, in der meist ohne Mitwirken eines Professors eine Gruppe von Studenten Lehrstoff selbst sucht und erarbeitet. Autonome Seminare stellen, insbesondere wenn die Prüfungsordnung die Vergabe von Scheinen zuläßt, eine alternative Lehrform zum sonstigen Universitätsbetrieb dar.

¹³Der Abschlußbericht des a_S D.O.M.E. ist für das ‘European Research Seminar on Advances in Distributed Systems’ als Vortragsvorschlag eingesandt worden, es liegt jedoch z. Z. noch keine Antwort des Veranstalters vor.

¹⁴siehe hierzu Bruckman:[18]

¹⁵siehe hierzu Curtis und Nichols: [5]

shutdown: ist ein Befehl, den man nicht benutzen sollte.
AMYLAAR, Elder wizard in TubMud
(Auszug aus dem Shutdown-Logfile)

Kapitel 4

Technische Konzepte

Entsprechend der Vielzahl unterschiedlicher Menschen, die Zeit in und mit MUDs verbringen, haben sich auch die MUDs selbst sehr unterschiedlich entwickelt in Hinsicht auf

- Ausgestaltung der Räume und Objekte des MUDs
- Umgang der Spielenden miteinander
- Verhältnis zwischen Leitung des MUDs und Spielenden
- und nicht zuletzt technischer Ausgestaltung.

Die im Lauf der Entstehungsgeschichte realisierten unterschiedlichen technischen Konzepte für die Implementierung von MUDs erlauben eine weitere Unterscheidung von MUDs und zwar nach technischer Ausgestaltung. Dabei sind die folgenden Punkte von Interesse:

- der verwendete Game Driver
- die Implementierungssprache für Erweiterungen der Datenbasis
- die Erweiterbarkeit an sich
- die verfügbaren Programmierbibliotheken

MUDs wurden für viele unterschiedliche Architekturen entwickelt, meist abhängig davon, welche Art von Computersystem dem jeweiligen Entwickler gerade zur Verfügung stand. Dementsprechend gibt es eine Vielzahl unterschiedlicher Implementierungen, teilweise mit eigens entwickelter Sprache zur Erweiterung der Datenbasis. Heute gibt für fast alle Rechner, auf denen eine Art von UNIX läuft, eine lauffähige MUD-Version, die direkt auf UNIX aufsetzt.

Eine Untersuchung und Katalogisierung der verschiedenen MUDs nach ihrer Implementierung soll jedoch nicht Gegenstand dieses Kapitels sein. Ich möchte vielmehr anhand von TubMud grundsätzlich die technische Realisierung eines MUDs erläutern.

4.1 Ein Blick hinter die Kulissen

Exemplarisch will ich nun am Beispiel TubMud einen Blick hinter die Kulisse der virtuellen Welt werfen. Während für die Spieler die Welt weitgehend so aussieht, wie in Kapitel 2 beschrieben, ist sie für Wizards, die ja programmieren und somit neue Objekte und Räume schaffen, eine andere. Ein Spieler, der Wizard werden möchte, und die nötige Anzahl von Erfahrungspunkten gesammelt hat, steigt im ersten Schritt zum Apprentice wizard, zum Wizardanwärter, auf. Zu diesem Zeitpunkt steht ihm oder ihr nur ein Teil des Befehlsumfangs der Wizards zur Verfügung (siehe unten). In einem zweiten Schritt wird dem Wizardanwärter ein Mentor zur Seite gestellt, der sowohl Hilfe in technischen Belangen bieten, als auch mit Rat und Tat zur Seite stehen soll.

Erst nach dem zweiten Schritt, in dem sich der Apprentice wizard seinen Mentor unter den Wizards gewählt und dieser per Zustimmung seine Aufgabe angenommen hat, steht dem Jungwizard der volle Befehlsumfang der Wizards zur Verfügung. Der Mentor ist für die Einführung des neuen Wizards verantwortlich. Er oder sie soll dem Neuling Einblick in die Programmierung des Muds geben, die Wizards' Rules einschärfen und darauf achten, daß der Jungwizard mit den neu gewonnenen Kräften ‚nicht über die Stränge schlägt‘.

Zu den neu gewonnenen Befehlen steht ein Hilfetext, der mit dem Befehl `help wizard` abgerufen werden kann, zur Verfügung:

```
> help wizard
Do 'cat /doc/w/<topic>' for more information.
```

Wizard commands:

ls <path>	cat <path>	load <file>	destruct <object>
ed <path>	clone <obj>	update <file>	rm <file>
mkdir <path>	cd <path>	more <path>	pwd
echo	echoall	domains	

Apprentice wizard commands:

wizlist	stat <name>	heal <name>	title <string>	align <string>
goto <obj>	in <obj> <cmd>	emote <msg>	people	
setmin <msg>	setmout <msg>	setmmin <msg>	setmmout <msg>	
review	shutdown	trans	force <player> to <cmd>	
won <chan>	woff <chan>	channels	wiz<chan> <msg>	

General info about creating rooms and objects: Look in `/doc/build`

Die im Hilfetext aufgeführten Befehle teilen sich zum einen in die zur Programmierung und Dateiverwaltung und zum anderen in die zur Objektmanipulation. Zusätzlich gibt es Hinweise auf Dokumentationstexte mit weitergehenden Informationen. Da viele der Spielenden, die Wizard werden, nur wenig Programmiererfahrung besitzen, gibt es in den meisten MUDs eine mehr oder weniger geordnete Menge von einführenden Texten, die den Umgang mit dem MUD als Programmierumgebung erleichtern sollen.

Erwähnenswert ist an dieser Stelle, daß Wizards eine eigene Kommunikationsmöglichkeit zur Verfügung steht, die den Spielern verborgen ist. Wizards können sich zu Kommu-

nikationskanälen ‚zuschalten‘, auf denen sie Nachrichten absetzen können, die von allen anderen zugeschalteten Wizards gehört werden. So gibt es zum Beispiel einen Kanal, auf dem Fragen zu Programmierproblemen gestellt werden können und einen zum chatten („sich unterhalten“), auf dem häufig Spielerausrufe und -aktionen kommentiert werden, zuweilen aber auch heftig gestritten wird.

4.1.1 Objekte, Räume — alles ‚nur Dateien‘ ...

Für jeden Wizard wird innerhalb des Mud-Dateisystems ein eigenes Verzeichnis angelegt, in dem Programme und andere Dateien gesichert werden können. Obwohl sich die Eingabeaufforderung für den Wizard nicht verändert hat, können jetzt auch Befehle zum Anzeigen von Dateien und Verzeichnissen eingegeben werden. Die Befehle, die mit der Dateiverwaltung zu tun haben, sind ähnlich wie unter Unix zu handhaben, um den Umgang damit möglichst einfach zu gestalten. Dateien können innerhalb des Muds angelegt, editiert oder gelöscht werden. Unter dem eigenen Verzeichnis kann der jeweilige Wizard Unterverzeichnisse anlegen oder auch wieder löschen. Wem es zu mühsam ist, Programme innerhalb des Muds mit dem eher sparsam ausgestatteten Editor `ed` zu schreiben, kann dies auch auf Betriebssystemebene mit dem eigenen Lieblingseditor tun und dann mit Hilfe der Übertragungsprotokolle `ftp` oder `mtp` die frisch programmierten Objekte ins Mud übertragen¹. Allerdings können die Dateien auch auf diesem Wege nur in den eigenen oder gemeinschaftlich benutzten Verzeichnissen abgelegt werden.

Grundsätzlich kann jeder Wizard jedes Programm innerhalb des Muds ansehen. Besonders nützlich ist dies für Jungwizards, die die Programmierung von Objekten für das Mud gerade lernen und so bei erfahreneren Programmierern nachschauen können, wie bestimmte Programmierprobleme gelöst werden. Gleichzeitig ergibt sich mit diesem Leserecht² auch eine gewisse Transparenz, die das Gefühl einer Gemeinschaft unter den Wizards schafft. Man hilft sich untereinander und gemeinsames Ziel ist es, das Mud lauffähig zu erhalten, es zu erweitern und für Spieler und Wizards gleichermaßen interessant zu gestalten.

Aus Systemsicherheitsgründen ist es nur für Elder Wizards, Archwizards und für Götter³ möglich, auf fremde oder zum System gehörende Verzeichnisse zuzugreifen und Dateien zu manipulieren. Zum einen besteht die Gefahr, daß auf systemeigene Objekte zugegriffen wird und dadurch das ganze Mud abstürzt. Zum anderen soll damit verhindert werden, daß allzu viele Wizards die Bibliotheksfunktionen verändern und diese inkonsistent werden. Wenn Funktionen, die in vielen Programmen verwendet werden, plötzlich verändert werden, kann es passieren, das bisher stabile Programme nicht mehr laufen und Spieler behindert werden, indem sie zum Beispiel einen Raum nicht mehr verlassen können. Kurz gesagt: ‚Viele Köche verderben den Brei‘. Bei den oben genannten Wizards höheren Ranges wird stillschweigend davon ausgegangen, daß sie über ausreichend Programmiererfahrung verfügen, um keine ernsthaften Fehler zu verursachen oder ihre Finger aus der Bibliothek lassen, wenn sie wenig oder gar nicht programmieren können.

Mit der Zeit ändert sich die Sicht eines Wizards auf das Mud durch das neu gewonnene

¹Zu objektorientierter Programmierung siehe [28].

²Dies wird nicht in allen MUDs so offen gehandhabt. In vielen MUDs kann kein Wizard Dateien anderer Wizards lesen.

³Ich gehe in Kapitel 5 noch näher auf die Hierarchiestrukturen ein.

Hintergrundwissen. Es ist nicht mehr nur eine virtuelle Welt, sondern hinter den Räumen und Objekten steht immer auch das entsprechende Programm. Betritt ein Wizard zum Beispiel einen Raum, so wird ihm noch vor der Beschreibung des Raumes der Dateipfad des Raumobjektes angezeigt. Bei der Kirche, in der sich im Kapitel 2 die Spielerin Jerdian aufhält, sieht das zum Beispiel so aus:

/room/church

You feel full of awe as you meet Hermes.

The portal of the village church is wide open. Dozens of tiny lights illuminate a dome at the far end of the church. The tower of the church lies east of the entrance. The place in front of the church is bordered by a pit to the west and the main road, crossing the meadow to the south.

A note is pinned to the signpost beside the portal.

There are two obvious exits: south and north.

Jerdian the witch (nice)

Hermes the messenger of the gods.

Die Kirche ist für Wizards nicht mehr nur ein Raum im Sinn der virtuellen Welt, sondern steht auch für ein Raumobjekt. Die Welt ist nun nicht mehr eine gegebene, sondern eine veränderbare, auf deren Gestaltung durch Programmierung Einfluß genommen werden kann.

4.1.2 Magie für Anfänger

Wie man dem Hilfetext entnehmen kann, gibt es Befehle zur Manipulation von Objekten. Mit dem Befehl `clone /obj/torch` wird das Objekt ‚Fackel‘ geschaffen, indem das gleichnamige Programm im Verzeichnis `/obj` aufgerufen wird. Die Fackel kann nun angesehen, weitergeben und angezündet werden. Mit dem Befehl `destruct torch` wird das Objekt wieder zerstört, also der Speicherinhalt freigegeben. Diese technischen Vorgänge sind vor den normalen Spielern hinter der Fassade der Magie versteckt. Jeder Spieler oder Wizard, der neben Sharilla steht, wenn sie eine Fackel klonet, erhält die Ausgabe:

Sharilla fetches something from another dimension.

Zerstört Sharilla die Fackel, sehen die anderen:

A torch is hurled into the void by Sharilla.

Ähnlich werden auch zuweilen auftretende Fehler vor den Spielern verborgen. Damit der Spieler merkt, daß etwas nicht in Ordnung ist und womöglich nicht stundenlang versucht, ein fehlerhaftes Objekt zu benutzen, gibt es zwar eine Fehlermeldung, diese ist aber ganz im Sinne der magieerfüllten Welt gehalten:

Your sensitive mind notices a wrongness in the fabric of space.

Wizards bekommen technische und detaillierte Fehlermeldungen, die zum Suchen und Beseitigen von Fehlern ja auch nötig sind.

Während die Spieler meist einen bestimmten Ort aufsuchen, indem sie alle anderen Räume auf dem Weg dorthin ‚zu Fuß‘ durchqueren, können Wizards einen Raum mit dem Befehl `goto` und dem Dateinamen des Raumobjektes direkt betreten oder verlassen. Vor dem Deckmäntelchen der Magie sieht das beim Betreten so:

Sharilla steps out of your shadow.

und beim Verlassen so aus:

Sharilla melts with her shadow and is gone.

Da Räume, Spieler und Dinge letztendlich alles nur Objekte sind, kann ein Wizard auch alle diese Objekte manipulieren. So kann zum Beispiel ein Spieler in einen anderen Raum befördert oder sein Lebensenergiewert verändert werden.

4.2 Der Game Driver

Der **Game Driver** ist **das** zentrale Programm, das das interaktive Spiel ermöglicht. Er ist vollständig in der Programmiersprache C ⁴implementiert und besteht in der Hauptsache aus den in Abbildung 4.1 zu sehenden Komponenten. Alle für das MUD geschriebenen Programme werden zuerst vom Compiler in eine interpretierbare Zwischensprache (Bytecode) übersetzt und dann, bei der Ausführung, vom Interpreter ausgeführt. Bei der Ausführung wechselt der Game Driver zwischen allen im MUD in Ausführung befindlichen Objekten. Das heißt, wenn er mit der Ausführung des Fackelanzündens beschäftigt ist, ruhen derweil alle anderen Aktivitäten anderer Spieler. Es findet eine rein sequentielle Abarbeitung der Aktivitäten statt. Jedes Objekt hat prinzipiell den Game Driver solange es will. Um zu vermeiden, daß fehlerhafte Objekte den Game Driver nicht mehr freigeben, gibt es einen Maximalwert an Ausführungszeit. Wird dieser überschritten, geht der Game Driver von einem Fehler im Objekt aus und bricht die Ausführung ab.

Für die regelmäßige Ausführung von Aktivitäten stellt der Game Driver unter anderem den Mechanismus des **heartbeat** zur Verfügung. Objekte können den heartbeat bei Bedarf anschalten. Auf diese Weise werden zum Beispiel Kämpfe zwischen Spielern und **Monstern** realisiert, in denen bis zum Tode eines Kontrahenten (bzw. vorzeitiger Beendigung durch Flucht) Kampfaktivitäten ausgeführt werden sollen. Da der Game Driver ebenfalls für das Bewegen der Objekte zuständig ist, würde er das Spielerobjekt im Fluchtfalle einem anderen Raumobjekt zuordnen, ihn sozusagen in einen anderen Raum bringen. Das Bewegen von Objekten ist eine relativ schwierige Operation, auf die ich noch näher eingehen werde.

Der Game Driver wurde als **Stackmaschine**⁵ konzipiert und besteht aus einem **Kontroll-** und einem **Wertestack**. Auf dem Kontrollstack werden Rücksprungadressen, Rahmenzeiger und ähnliches abgelegt, der Wertestack wird für die Auswertung von Ausdrücken, für lokale Variablen und für Funktionsargumente benutzt. Dabei ist zu bemerken, daß Argumente und lokale Variablen gleich behandelt werden ⁶ Beim Funktionsaufruf werden die

⁴Zur Programmiersprache C siehe: [21]

⁵Zur Stackmaschine siehe: [29]

⁶siehe [26], Kapitel 16

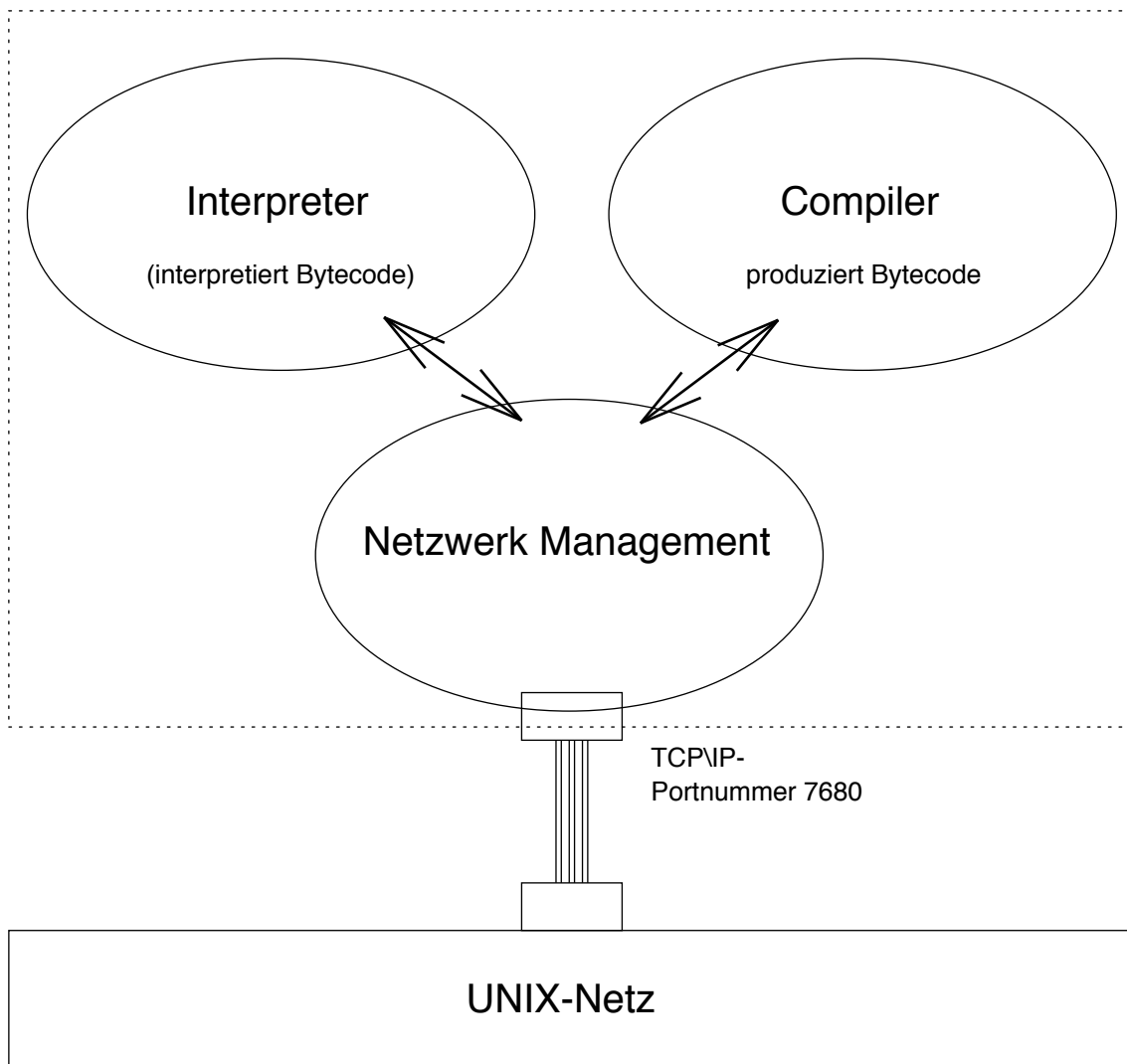


Abbildung 4.1: Bestandteile des Game Drivers

Argumente ausgewertet und auf dem Wertestack abgelegt. Die Anzahl der Argumente wird dabei auf dem Kontrollstack vermerkt. Auf diese Art und Weise kann mittels des Rahmenzeigers, der auf den Anfang der Argumente bzw. lokalen Variablen einer Funktion zeigt, schnell auf die benötigten Werte zugegriffen werden.

Die LPMUD-Driver, zu denen auch der von TubMud⁷ gehört, zeichnen sich gegenüber früheren Entwicklungen anderer MUDs durch verschiedene vorteilhafte Eigenschaften aus:

- ‚on the fly object loading‘
- ‚rollende‘ Resets
- Interpretation vorcompilierter Codes (ab Version 2.0)
- Vererbung (ab Version 2.4)
- Realisierung als Stackmaschine (ab Version 3.0)
- Mehrfachvererbung (ab Version 3.0)
- virtuelle Vererbung (ab Version 3.0)
- Lambda Closures und andere komplexe Datentypen (ab Version 3.1)

Wie ich bereits im Kapitel 3 beschrieben habe, versuchte Lars Pensjö die Vorteile unterschiedlicher Game Driver zu vereinen, als er um 1984 den ersten LPMUD-Driver implementierte. Hier muß an erster Stelle die Möglichkeit, zur Laufzeit des Game Drivers Objekte zu laden, genannt werden. In früheren Implementierungen von MUDs war es häufig nötig, das ganze Spiel neu zu übersetzen und zu starten, wenn ein neues Objekt eingebunden werden sollte, da alle Objekte des MUDs mit im Game Driver eingebunden waren. Dies hatte zur Folge, daß frühere MUDs lange Zeiten hatten, während derer nicht gespielt werden konnte, weil das MUD erweitert oder von Fehlern befreit wurde. Es war üblich, das frühe MUDs Öffnungszeiten hatten, während derer gespielt werden konnte und dann wieder Wartungsarbeiten vorgenommen wurden. Teilweise gab es auch ein MUD, in dem gespielt werden konnte und eines, das dem Entwickeln neuer Programme vorbehalten war.

Die Teilung des Game Drivers in Compiler und Interpreter ermöglicht es, die Einbindung neuer Objekte quasi auszulagern. Das Programm eines neuen Objektes wird erst vom Compiler übersetzt. Dazu wird genau für diesen Vorgang der Compiler gestartet. Compiler und Interpreter arbeiten unabhängig voneinander. Das Compilieren neuer Programme ist für den Interpreter nicht von Belang, er interpretiert jeden ihm zur Verfügung stehenden Zwischencode. Auf diese Weise kann nicht nur während der Erweiterung des MUDs gespielt werden, sondern es ergibt sich auch ein großer Zeitgewinn bei der Programmierung, da mehrere Programmierer gewissermaßen gleichzeitig arbeiten können.

Eine weitere Erleichterung sind die rollenden Resets. Während es lange Zeit üblich war, regelmäßig alle Spieler an einen Ort zu rufen und das ganze MUD zu resetten (alle Objekte in den Anfangszustand zu bringen), bevor weitergespielt werden konnte, werden in den LPMUDs alle Objekte nacheinander in regelmäßigen Zeitabständen rückgesetzt ohne daß der Spielfluß unterbrochen wird.

⁷Momentan läuft TubMud mit der Game Driver-Version 03.02.1@40 des Entwicklers Jörn Rennecke (Amylaar@TubMud).

Während frühe MUDs, wie MUD1 von Trubshaw und Bartle noch in C programmiert waren, wurden im Lauf der Zeit die Konzepte objektorientierter Programmierung genutzt und in eigens für das jeweilige MUD ausgearbeitete Programmiersprachen umgesetzt. Beispiele dafür sind LambdaMOO von Pavel Curtis und die LPMUDs, für die Lars Pensjö LPC (Lars-Pensjö-C) entwickelt hat, auf das ich im Kapitel 4.3.1 noch näher eingehen werde.

4.3 Eine objektorientierte Welt

Während der Game Driver in C implementiert ist, werden alle Objekte innerhalb des MUDs in LPC realisiert. Bei LPC handelt es sich um eine von Lars Pensjö und anderen entwickelte objektorientierte Programmiersprache mit C-ähnlicher Syntax. LPC verfügt wie schon erwähnt über Mehrfachvererbung und ein striktes Typkonzept, das allerdings für Funktionsprototypen auch abgeschaltet werden kann. Ein LPC-Programm kann aus vier Komponenten bestehen:

- Anweisungen von welchen Klassen geerbt werden soll
- Präprozessoranweisungen
- Variablen
- Funktionen

Ich werde mich in den nun folgenden Teilen, die einen kurzen Einblick in LPC, die Bibliotheksfunktionen der MudLib und einige typische Programmiermethoden geben, exemplarisch mit TubMud auseinandersetzen, da ich über dieses MUD die weitestgehenden Informationen habe. Genauere Informationen zu vielen von mir erwähnten Funktionen finden sich im LPmud-Manual⁸

4.3.1 Ein Einblick in LPC

Wie bereits erwähnt, sind die Teile des MUDs, die nicht zum Game Driver gehören, sämtlich Objekte, egal ob es sich nun um einen Raum, ein Monster, einen NPC oder eine Fackel handelt. Die Objekte unterscheiden sich allerdings hinsichtlich ihrer Verwendung und ihres Verhaltens. Ich werde diese Unterschiede im Folgenden näher betrachten.

Im Gegensatz zu objektorientierten Programmiersprachen, wie zum Beispiel Smalltalk, ist in LPC das Konzept der Mehrfachvererbung realisiert. Das heißt, daß eine neue Objektklasse verschiedene andere Klassen erben und die neuen Objekte somit gleichsam deren Funktionen benutzen können. Dabei sind Objekte die Instanzierungen der jeweiligen Klasse. Beispielsweise kann Sharilla mit dem clone-Befehl eine Fackel erzeugen, die dann eine Instanz der neuen Klasse `/complex/lamp` ist, die die Klassen `/basic/light` und `/complex/item` erbt. Alle diese Klassen werden durch eine Datei beschrieben. Bei

⁸siehe hierzu: LP-MUD-Manual [26]. Außerdem befindet sich im LP-MUD-Manual ein ‘Agreement’ genannter Abschnitt zu den Lizenzrechten an für TubMud geschriebenen Programmen.

Einfachvererbung kann ein Objekt nur von einer einzigen anderen Klasse Variablen und Funktionen erben, die dann aber genau wie bei der Mehrfachvererbung wie eigene, selbst-deklarierte benutzt werden können. Dies hat den Vorteil, daß nicht die Gefahr besteht, daß sich Variablen oder Funktionen gegenseitig überlagern, weil sie in mehreren Objekten mit gleichem Namen deklariert sind. Bei LPC kann laut Manual von maximal 60 Klassen geerbt werden⁹. Da es für LPC kein Werkzeug zum Durchforschen der Klassenhierarchie gibt, wie es zum Beispiel die Entwicklungsumgebung von Smalltalk es vorsieht, ist die im Laufe der Zeit angewachsene Menge von bereits vorhandenen Klassen besonders für ungeübte Programmierer sehr unübersichtlich, was die Suche nach Programmierfehlern immer wieder zu einer Herausforderung werden läßt.

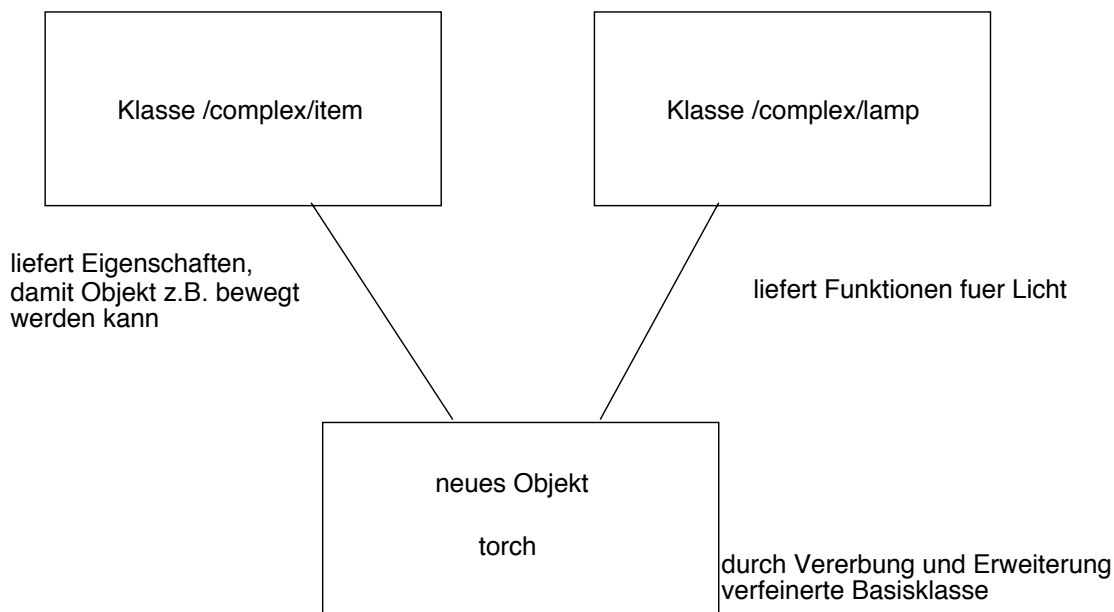


Abbildung 4.2: Ein Beispiel für Mehrfachvererbung in LPC.

4.3.2 Objekte und Objekte und Objekte...

Die Mudlib stellt alle schon vorhandenen Klassen zur Verfügung. Dabei handelt es sich sowohl um Basisklassen, wie /complex/item, die nur eine sehr allgemeine Basis für die Erstellung einer neuen Objektklasse sind und noch verfeinert werden können, als auch alle Objekte, die andere Wizards programmiert haben und die unter Umständen schon sehr genau ausgearbeitet sind. Manche der Klassen, die noch nicht verfeinert sind, können auch erst ge'cloned und dann konfiguriert werden, andere sind schon fertig einsetzbar. Der Vorgang des Klonens umfaßt mehrere Arbeitsschritte:

- Laden des Programms und

⁹siehe LPmud-Manual:[26], Seite 30

- Übersetzen in den Zwischencode, wenn es bisher noch nicht geladen war
- Interpretieren des Zwischencodes und Erschaffen eines ‚Originals‘ (blueprint)
- Erschaffen eines Klons als Instanziierung des Originals

Beim Klon-Versuch werden dann auch mögliche Fehlermeldungen an den Wizard ausgegeben. Will er nach Verbesserung der Fehler das Objekt erneut klonen, so müssen erst alle Klone der vorherigen Version aus dem Speicher gelöscht, sowie das Original überschrieben werden. Dieses System von Original und Klone hat den Vorteil, daß der Zwischencode jedes Objekts nur einmal geladen und im Speicher gehalten wird. Dadurch hat das MUD einen geringeren Speicherverbrauch. Das Original enthält den Programmcode, der von allen Instanzen gemeinsam benutzt wird. Es wird nicht gelöscht, auch wenn der letzte Klon zerstört ist. Es könnte ja sein, daß zu einem späteren Zeitpunkt ein anderer Spieler ein solches Objekt benötigt – Fackeln werden zum Beispiel ziemlich häufig gebraucht und es wäre ungünstig, wenn jedes Mal das Programm neu geladen und übersetzt werden müßte. Das Originalobjekt wird erst mit dem nächsten Neustart des gesamten MUD vernichtet. Danach werden alle Objekte in genau dem Moment geladen, wenn sie von einem Spieler angefordert werden. Objekte können im Spielverlauf auch von anderen Objekten geklont werden. Im allgemeinen wird ein Wizard ein Objekt nur im Versuchsstadium selbst klonen.

Obwohl letztendlich alle Objekte sind, kann man sie doch nach der Art der Verwendung unterscheiden in:

- Raumobjekte
- Monsterobjekt und NPCs
- Playerobjekte
- sonstige Objekte

Raumobjekte unterscheiden sich von den übrigen dadurch, daß von ihnen meist nur das Original existiert. Da man sowieso keine zwei identischen Räume haben will, wird bei Räumen dem Spieler oder Wizard direkt das Original zugeordnet, das dann betreten werden kann. Monster und NPCs werden beim Klonen wie oben beschrieben behandelt. Monster, NPCs und Playerobjekte haben das Attribut `living`. Das heißt, daß bei ihnen die ererbte und im Game Driver verankerte Bibliotheksfunktion `enable_commands()` aufgerufen wurde. Dies ist für alle Objekte nötig, die mit anderen interagieren können sollen und daher eine Kommandoliste brauchen¹⁰. Sie können dann alle Kommandos, die normalerweise für Spieler zugänglich sind¹¹, verwenden. Für jedes Objekt mit dem Attribut `living` verwaltet der Game Driver eine `sentence-list`. In dieser Liste stehen die Kommandos, die das `living`-Objekt aufrufen kann und ein Verweis auf das Objekt und die Funktion, die das Kommando repräsentiert, und die dann gegebenenfalls aufgerufen wird. Loggt sich beispielsweise ein Spieler ein, so ist seine `sentence-list` initial leer. Es werden dann aber

¹⁰Siehe Zeichnung 4.3.

¹¹siehe auch Hilfstext in Kapitel 2

sogleich die im Programm `player.c` implementierten Kommandos/Funktionen eingetragen. So stehen dem Spieler direkt nach dem Einloggen die Grundkommandos `say`, `score`, `quit` usw. zur Verfügung. Im Gegensatz zu Monstern und NPCs können Playerobjekte aus Systemsicherheitsgründen in konfiguriertem Zustand, also als fertiger Spieler, nicht von Wizards oder anderen Objekten geklont werden. Dies ist dem Masterobjekt des Game Drivers beim Einloggengang vorbehalten, da ein bereits konfiguriertes Playerobjekt ja einem Spielenden „gehört“ und nur ihm vorbehalten sein soll.

Alle übrigen, sonstigen Objekte, also zum Beispiel Fackeln, Waffen, Kleidung, ganze Blumensträuße oder dergleichen sind — so nicht per Programm selbst verhindert — frei klonbar. Der Wizard hat sie automatisch in seinem Zugriffsbereich, seinem `Inventory`. Da sich Objekte beliebig tief verschachtelt ineinander befinden können (wobei allerdings Ringschlüsse verhindert werden), muß von jedem Objekt bekannt sein, in welchem anderen es sich befindet und welche Objekte ihrerseits in ihm sind. Befindet sich ein Spieler in einem Raum, zum Beispiel der Kirche, so ist er im `Inventory` des Raumobjekts Kirche eingetragen. Das Spielerobjekt hat seinerseits ein `Environment`, in dem festgehalten ist, daß er der Kirche zugeordnet ist, sich also in ihr befindet. Nun kann der Spieler einen Beutel mit sich führen. Der Beutel ist dann im `Inventory` des Spielers eingetragen und hat als `Environment` den Spieler. Die im Beutel befindlichen Objekte sind im `Inventory` des Beutels eingetragen. . .

Auf diese Weise kann jedes Objekt genau lokalisiert und damit über die Kommando-schnittstelle festgestellt werden, ob es mit einem anderen jeweils interagieren kann.

4.3.3 Kissen sind zum Werfen da!

Lokale Funktionen von Objekten werden `Lfuns` genannt. Sie können intern verwendet werden, aber auch, um anderen Objekten die Interaktion mit dem eigenen Objekt zu ermöglichen. Zum Beispiel stellen Objekte, wie eine Fackel oder ein Kissen Kommandos für `living`-Objekte zur Verfügung. Diese lokalen Funktionen sind im allgemeinen über den Mechanismus der `add_action()`¹² realisiert. In der `sentence-list` des Spielers wird das Kommando, mit dem die Funktion aktiviert wird, eingetragen und ein Verweis auf die Implementierung der Funktion im Kissenobjekt (Kissen-werfen zum Beispiel). `Lfuns` von Objekten wie Kissen, Fackeln oder dergleichen werden dann eingetragen, wenn sich das Objekt in Reichweite des `living`-Objektes befindet, er es also in der Hand hält (`Inventory`) oder es greifbar ist (`Environment`). `Lfuns` von Monstern oder NPCs werden dann eingetragen, wenn sich beide im gleichen `Environment` befinden. Auf diese Art und Weise können Spieler und Monster oder auch NPCs unter sich interagieren. Zeichnung 4.3 verdeutlicht diese Zusammenhänge.

Bewegt sich ein Objekt von einem Raum in einen anderen, so müssen eine Menge Dinge bedacht werden. Zuerst muß geprüft werden, ob unser Objekt dem neuen Raumobjekt überhaupt zugeordnet werden kann. Dies sollte bei verschlossenen Türen oder wenn das neue Raumobjekt zu klein ist, zum Beispiel nicht gehen). Außerdem soll verhindert werden, daß ein erstes Objekt in ein zweites und dann das zweite in das erste bewegt wird, was schlicht und ergreifend unlogisch wäre. Danach wird das `Licht-Level` für unser Objekt

¹²siehe Programmbeispiel im Anhang

neu gesetzt (geht man aus einem hellen Raum in einen dunklen, so kann man den Raum nicht mehr überblicken, selbst wenn man ein Monster ist). Weiterhin wird geprüft, ob das Objekt frisch geladen oder geklont wurde und so mit noch kein Environment hat. Hatte es bereits ein Environment, so muß jetzt die sentence-list angepaßt werden, falls es sich unser Objekt living war. In diesem Fall müssen alle Kommandos, die vom alten Raum oder Objekten in ihm oder unserem eigenen Objekt in bestimmten Räumen (zum Beispiel könnte ja ein Hund prinzipiell ein Bellkommando aktiviert haben, wenn er sich in geschlossenen Räumen befindet) zur Verfügung gestellt wurden, aus der sentence-list entfernt werden. Außerdem müssen alle anderen, im alten Raum befindlichen Objekte ihrerseits auf Kommandos überprüft werden, die unter Umständen durch unser wanderwilliges Objekt aktiviert wurden. Nötigenfalls müssen alle anderen Objekte auf den neuen Stand gebracht werden. Zusätzlich werden im alten und neuen Environment Funktionen aufgerufen, die eine Benachrichtigung für sich in den Räumen befindende Objekte darstellen. Damit kann auf die Bewegung des Player-Objektes reagiert werden, indem zum Beispiel bei Betreten eines Raumes das Fenster durch den Luftzug zuschlägt. **Nun** endlich kann unser Wanderobjekt dem neuen Raum zugeordnet und in dessen Inventory eingetragen werden. Das neue Environment unseres Objektes ist nunmehr der Zielraum.

Die Veränderung der Umgebung eines Objektes oder das Auftauchen eines bestimmten Objektes kann wie schon angedeutet Auswirkungen auf andere sich im neuen Raum befindliche Objekte oder auf sich selbst haben, außerdem muß unter Umständen die sentence-list aktualisiert werden. Daher wird nun bei jedem sich im neuen Raum befindlichen Objekt (jedes Objekt im Inventory des neuen Raumes) die Lfun `init()` aufgerufen. In Ihr sind mögliche Reaktionen auf unser Objekt, sowie im allgemeinen die `add_actions` realisiert. Außerdem wird in unserem Objekt für jedes living-Objekt im neuen Raum ebenfalls einmal die Funktion `init()` aktiviert. Auf diese Weise werden Reaktionen von Objekten aufeinander und die Aktualisierung der jeweiligen sentence-lists verwirklicht.

Mit dem Mechanismus des `call_other` kann ein Objekt Lfuns eines anderen Objektes auslösen bzw. aufrufen. Dies ist neben der `tell_object()`- und `catch_tell()`-schnittstelle die einzige Möglichkeit der technischen Inter-Objekt-Kommunikation in LPC.

Neben den lokalen gibt es noch die externen Funktionen, die Efuns. Dabei handelt es sich um globale, im Driver in C implementierte Basisfunktionen, die jedes Objekt unter Umständen einmal braucht und die meist schwer oder gar nicht in der Programmiersprache selbst zu realisieren sind. Ein Beispiel ist die Funktion `explode`, mit der ein Text-String in ein Feld von einzelnen Wörtern umgewandelt werden kann. Auf efuns kann jedes Objekt zugreifen, sie können aber von Lfuns überlagert werden, woraus sich bei geeignet unüberlegter Ererbung von Funktionen interessante Fehler und Probleme ergeben können. Weitere wichtige efuns sind `add_action()` und `write()` für die Kommandoerstellung und Textausgabe.

4.3.4 LPC für Fortgeschrittene

Bis hier gekommen, muß man eigentlich zuzugeben, daß LPC eine etwas unorthodoxe Mischung verschiedener Programmierkonzepte ist, die nichts desto trotz ihren Zweck, nämlich die einfache Erweiterung der MUD-Datenbasis, hervorragend erfüllt. Ein applikatives Element von LPC sind die sogenannten Lambda-Closures, die bei Lisp oder

verwandten Programmiersprachen entliehen sind. Closures existieren neben den imperativen restlichen Mechanismen von LPC und erlauben die Erstellung von dynamischem Code. Im einfachsten Fall besteht die Closure nur aus dem Aufruf einer Efun, es können aber auch umfangreichere Funktionen ‚zusammengebastelt‘ werden. Zum Beispiel kann man auf elegante und kurze Weise Funktionen schreiben, die es dem Spieler erlauben, über eine `add_action()` dynamisch angepasste Reaktionen zu bewirken. Wollen wir dem Spieler beispielsweise eine Katze vorsetzen, die zur Tageszeit getigert, nachts jedoch grau ist (wie alle anständigen Katzen), so erreichen wir dies am schnellsten mittels einer Closure. Closure-Code wird schneller als üblicher LPC-Code ausgeführt, da bei mehrfacher Ausführung des gleichen Codes Interpreter-Overhead wegfällt, der sich natürlich auf die Ausführungsgeschwindigkeit auswirken würde.

Aber zurück zu unserer Katze. Jedes Objekt hat eine Kurzbeschreibung (festgelegt mit `set_short()`), die zum Beispiel ausgegeben wird, wenn sich ein Spieler mit dem befehl `look` im Raum umschaute. Die Kurzbeschreibung wird einer Variable aus `/basic/description` übergeben und dort für jedes Objekt privat gehalten. In unserem Fall also privat für die Katze und nicht ohne weiteres für andere Objekte manipulierbar. Soll die Katze also ein variables Aussehen haben, so muss sie gewissermaßen selbst dafür sorgen. Mit der Funktion `set_short()` wird normalerweise ein fester Text übergeben. In diesem Fall ersetzen wir ihn durch eine Closure, die dann die mudweit gerechnete Tageszeit ermitteln und in Abhängigkeit davon einen entsprechenden Text wählen muß. Ob ein Text oder eine Closure übergeben wurde, stellt die Funktion `query_short()` fest. Wenn es sich um eine Closure handelte, wird diese ausgeführt und das entsprechende Ergebnis dem Spieler ausgegeben.

4.4 An's Licht, zur Sonne...

Wie bereits mehrfach erwähnt, basiert die Erweiterbarkeit der LPMuds auf der Trennung von Game Driver und Mudlib. Die einzelnen Komponenten sind in Abbildung 4.1 verdeutlicht. Der Game Driver führt die Funktionen der Mudlib-Programme aus, compiliert neue Programme, ermöglicht das Einloggen der Spieler, stellt die Anbindung ans Internet her und bietet Automatismen, wie Resets und den Heartbeat an. Die Efunns sind vom Game Driver zur Verfügung gestellte Funktionen, also gewissermaßen die „Primitiven“ der Programmiersprache LPC. Die Mudlib selbst besteht aus den grundlegenden, einfachen Klassen wie `/complex/item` oder `/basic/light`. Außerdem gehören alle Verzeichnisse der Wizards dazu. Damit wird jede programmierte Klasse automatisch Teil der Mudlib.

Der Game Driver ermöglicht das Einloggen neuer Spieler vom Internet. Da er aus nur einem Prozeß besteht, wird die maximale Anzahl der Spieler auf jeden Fall durch die von Unix-Seite festgelegte maximale Anzahl von File-Deskriptoren festgelegt. Diese Zahl ist mit (meist) 64 aber in jedem Fall so groß, daß das Mud unerträglich langsam werden würde.

Will sich ein neuer Spieler mit dem Befehl `telnet morgen.cs.tu-berlin.de 7680` einloggen, wird vom telnet-Programm der Systemcall `connect()` ausgelöst. Der Game Driver trägt dann ein neues `struct interactive` in die von ihm verwaltete Liste ein und ruft die Funktion `connect()` im Masterobjekt des MUDs auf. Wenn eine Verbindung zum MUD angefragt

wird, führt der Game Driver ein `accept()` aus. Das Masterobjekt kloniert nun `/obj/login`, das dem struct `interactive` zugeordnet wird. `/obj/login` gibt den folgenden Begrüßungstext aus:

```
Welcome to TUBMUD (LPmud Amylaar 03.02.1@40)
Use the guest name if you just want a look.
```

```
Type: 1 - to create a new character,
      2 - to quit,
      3 - to call the random name generator,
or enter your name if you've already got a character.
```

Enter choice or name:

Der Spielende gibt nun Namen und password seines Spielers ein. `/obj/login` sorgt dafür, daß es selbst durch `/obj/player` ersetzt (überlagert) wird, das dann seinerseits wiederum dem struct `interactive` zugeordnet wird. Der Vorgang des Einloggens ist beendet und der Spieler hat das MUD betreten.

Eine Verdeutlichung des Einloggenvorgangs ergibt sich aus Abbildung 4.4:

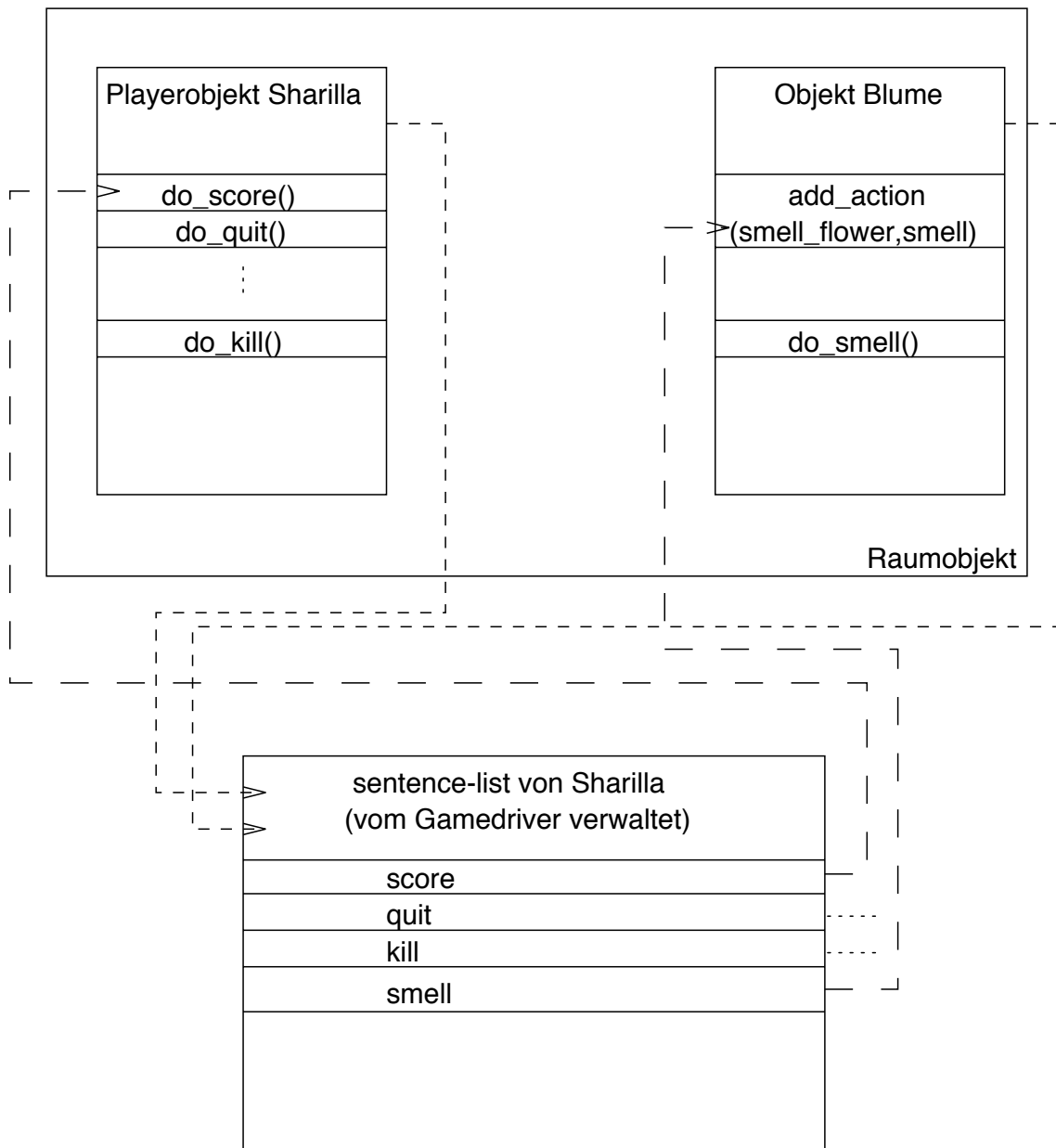


Abbildung 4.3: Zusammenwirken von Objekten

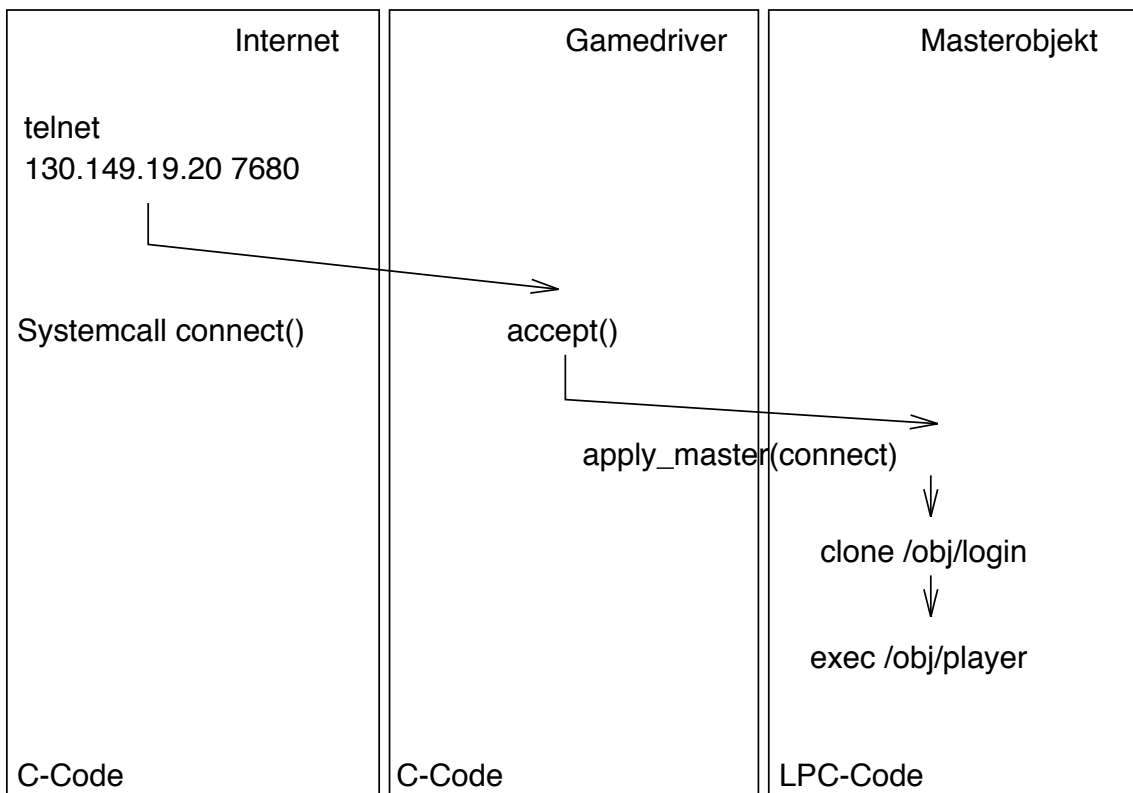


Abbildung 4.4: Vorgänge beim Einloggen

Kapitel 5

Gesellschaft en miniature

Unter den Autoren und Autorinnen, die sich in den vergangenen Jahren wissenschaftlich mit MUDs auseinandergesetzt haben, ist es unumstritten, daß sich im Lauf der Zeit innerhalb dieser virtuellen Welten eine eigene Gemeinschaft bildet¹ und fortwährend weiterentwickelt. In den mittlerweile knapp 2 Jahren, die ich zur virtuellen Gemeinschaft von TubMud zähle, habe auch ich diese Erfahrung gemacht und es ist genau diese **Gemeinschaft**, die das MUD für mich so faszinierend macht. Howard Rheingold benutzt den englischen Begriff ‚community‘, der ins Deutsche übertragen sowohl für „Gemeinschaft“, als auch „Gemeinde“ steht, was im Prinzip ja der englischen Benutzung entspricht. Im Deutschen gibt es jedoch Unterschiede zwischen den beiden Begriffen. Eine Gemeinschaft bildet sich durch Interessierte, die irgendetwas gemeinsam haben, man wird aktiv Mitglied einer Gemeinschaft. In eine Gemeinde hingegen wird man hineingeboren, ohne daß man selbst die Entscheidung treffen kann. Meiner Meinung nach trifft der deutsche Begriff „Gemeinschaft“ eher auf MUDs zu, weil man sich ein MUD, in dem man Zeit verbringt, ja aussucht. Daher werde ich mich auf diese Übersetzung beschränken.

Pavel Curtis spricht von MUDs als ‚social virtual reality‘², da sich nach seiner Beobachtung im Laufe der Zeit eine tatsächliche Gemeinschaft bildet, indem die Individuen der Gemeinschaft zu einem Konsens über gemeinsamen Sprachgebrauch, angemessenes Verhalten anderen gegenüber und die soziale Rolle öffentlicher Orte gelangen.

Howard Rheingold definiert in der Einleitung seines Buches virtuelle Gemeinschaften als „soziale Zusammenschlüsse, die dann im Netz entstehen, wenn genug Leute diese öffentlichen Diskussionen lange genug führen und dabei ihre Gefühle einbringen, so daß im Cyberspace ein Geflecht persönlicher Beziehungen entsteht“. Er betrachtet diese Gemeinschaften als „soziale Petrischale“, bei der das Netz die Nährlösung und die virtuellen Gemeinschaften selbst die Kolonien von Mikroorganismen sind. Ein „soziales Experiment, das zwar von niemand geplant wurde, aber trotzdem stattfindet“.

Im Gegensatz zu der prozeßorientierten Sicht von Curtis und der an den sozialen Bindungen orientierten Sicht von Rheingold, sieht Amy Bruckman MUDs vor allem als ‚identity workshop‘, als einen Freiraum, die eigene Persönlichkeit in einer „warmen sozialen Umgebung“ zu entdecken und zu entwickeln. Sie vertritt außerdem die Ansicht,

¹nachzulesen bei Bruckman[1], Kapitel 2.4, Curtis[4], Kapitel 2.3, Reid[3], Kapitel 1, Rheingold [2], Einleitung, Seite 16 ff. und anderen

²Kann hier übersetzt werden als „sozial betonte virtuelle Realität“.

daß man, wenn man sich länger mit MUDs beschäftigt, nicht umhin kommt, ein größeres Verständnis für soziale Phänomene zu entwickeln. Elizabeth Reid geht über diese Sicht hinaus, indem sie MUDs als „wichtige soziale Werkzeuge“ bezeichnet. Nach ihrer Darstellung bilden drei Arten von Objekten, „Orte, Dinge und Leute“ den Kontext, in dem die MUD-Gemeinschaft agiert. Dabei sind die reiche Kommunikation der Bewohner und das Erschaffen eines sozialen Kontextes die Basis dieser Gemeinschaft.

Ich werde mich im Folgenden mit diesen Thesen auseinandersetzen, indem ich den Aufbau und die Strukturen von TubMud einer näheren Betrachtung unterziehe. Es ist aber meine Meinung, daß sich in den MUDs nicht nur eine Gemeinschaft bildet, sondern darüber hinaus eine eigene kleine Gesellschaft, sozusagen ein sozialer Mikrokosmos, eine „Gesellschaft en miniature“. Nach meiner Beobachtung verselbständigen sich die Gemeinschaften in den virtuellen Welten und durch das Schaffen nicht nur eines gemeinsamen Kontextes, sondern richtiger gesellschaftlicher Strukturen, wie Gruppen, Interessenvertretungen und Herrschaftsstrukturen verselbständigen sich die Gemeinschaften zu virtuellen Gesellschaften. Teilweise ähneln sie den realen Gesellschaften in Strukturen und gesellschaftlichen Prozessen, teilweise sind sie aber auch ganz anders wie sich zeigen wird.

5.1 Aufbau des sozialen Mikrokosmos TubMud

Obwohl TubMud zumindest von seiner Programmbasis her ein abenteuerorientiertes Mud ist, fügt es sich doch nicht in die strenge Kategorisierung, die sowohl Reid als auch Rheingold vornehmen³. Es hat zwar Elemente eines typischen abenteuerorientierten MUDs, wie Kämpfe und zu lösende Aufgaben oder Rätsel, aber es hat auch ein ausgeprägtes soziales Gefüge, auf das ein neuer Spieler bereits kurz nach Betreten des MUDs hingewiesen wird⁴. TubMud muß also irgendwo zwischen den klassischen abenteuerorientierten und den sozialen MUDs angesiedelt werden.

Die Gesellschaft in TubMud teilt sich zu allererst einmal analog zu den in Kapitel 4 dargestellten technischen Unterschieden in die Gruppen der Wizards und Player. Wie auch in Abbildung 5.1 zu sehen, bildet die Gruppe der „einfachen“ Spieler die Basis der Gesellschaft, von der aus ein Aufstieg in die Gruppe der Wizards oder in die der Peers⁵ möglich ist.

5.1.1 Die Gruppe der Wizards

Die Strukturen innerhalb der Gruppe der Wizards existieren nicht nur länger, als die der Clans, sondern sind dadurch auch mehr in sich gefestigt und ähneln fast einer parlamenta-

³dazu: Rheingold[2] Einleitung und Seite 186 ff. und Reid[3] Seite 53 ff.

⁴Aus den im Anhang zu findenden Regeln für Spieler: „First: This is a GAME. Players are here to have fun. This is why you should treat everyone here as you would treat your friends in the REAL world. Misunderstandings and hurt feelings also occur in electronic communication and can be as unpleasant to all sides as in REAL life. The MUD is not a place where you can do anything you can't do in REAL life. It is a place to have fun with other people. Politeness, friendliness and helpfulness are a guarantee that we all can have fun here., und „This is more than just a game, because many of us don't just „play,“ here, but we also meet in small and large groups. We met here in the MUD and became friends in the REAL world.,“

⁵Peers sind Clanmitglieder höherer Ränge.

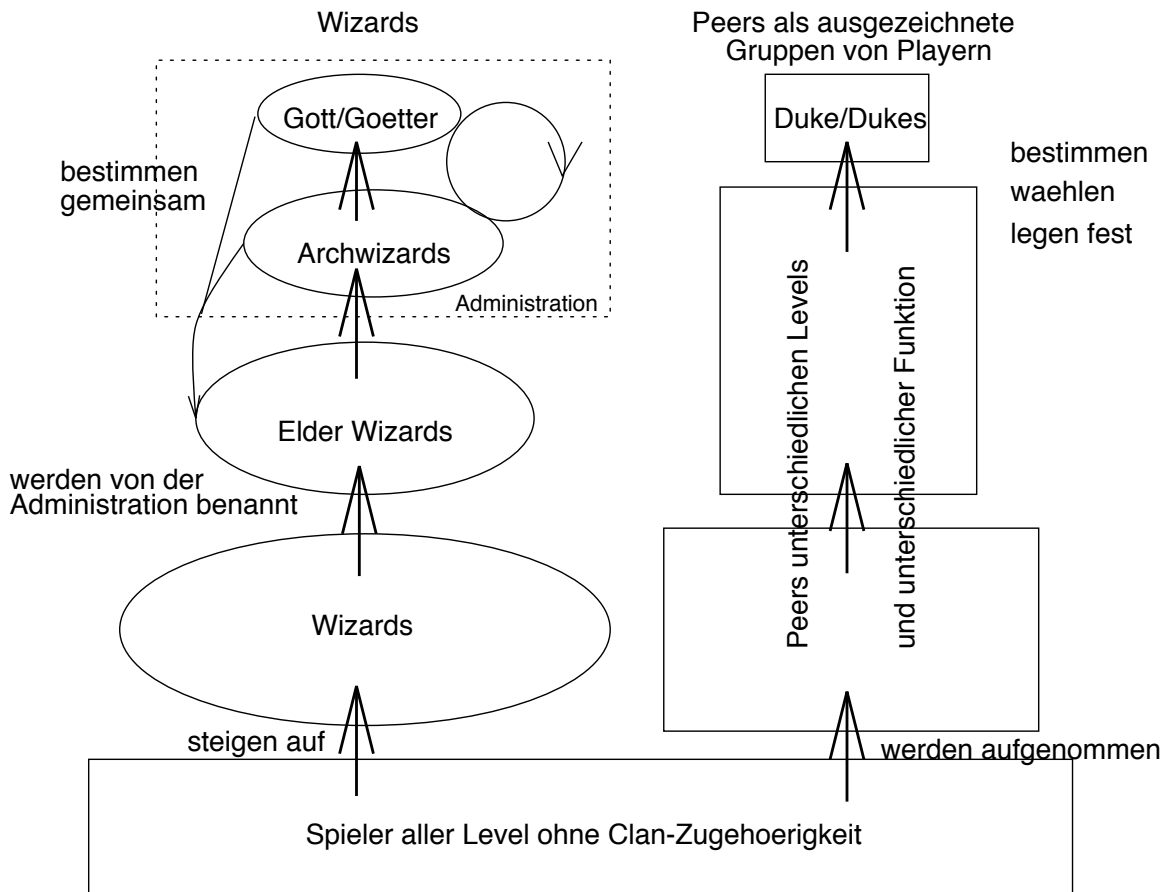


Abbildung 5.1: Aufbau der TubMud-Gesellschaft

rischen Regierung mit Verwaltung, was allerdings keine Gewähr dafür ist, daß es auch nach demokratischen Prinzipien zugeht. Der Aufstieg vom Spieler zum Wizard läuft noch fast automatisch nach Erreichen der nötigen Zahl von Erfahrungspunkten ab. Entscheidet sich ein Spielender dafür, Wizard zu werden, kann er danach eigene Objekte in beliebiger Anzahl programmieren. Eine Beschränkung bei der Nutzung von Ressourcen, wie zum Beispiel in LambdaMOO gibt es in TubMud nicht. Aus der Gruppe der Wizards werden von der Administration des MUDs, bestehend aus den Archwizards und dem Gott (siehe auch Abbildung 5.1) die Elder Wizards benannt. Der Titel eines Elder Wizard ist, wie mir Demos, der Gott von TubMud, in einem Gespräch bestätigte, ein reiner Ehrentitel, der Wizards für besondere Verdienste um das MUD verliehen wird. Da dies jedoch nicht schriftlich festgelegt ist, kommt es bezüglich der Position der Elder Wizards immer wieder zu Verwirrung, denn obwohl die Elder Wizards den Wizards nicht vorgesetzt sind, haben sie doch de facto besondere Rechte, die sie fast in die Position eines Vorgesetzten bringen und die von manchen Elder Wizards auch so interpretiert wird. Da es wie bereits gesagt, kaum schriftliche Festlegungen gibt, vermengen sich persönliche Vorstellungen mit Gewohnheitsrecht. Zum Beispiel werden auf der monatlich stattfindenden Elder-Conference nach meinungsbildender Diskussion wichtige Entscheidungen mit sowohl technischen als

auch sozialen Auswirkungen getroffen. An der Elder-Conference nehmen die Elder- und Archwizards redeberechtigt teil. Ursprünglich war sie als nicht-öffentliche Veranstaltung geplant, da man, wenn viele Leute gleichzeitig „reden“, mit dem Lesen der Wortbeiträge nicht schnell genug hinterherkommen kann.⁶ In letzter Zeit nehmen vermehrt „normale“ Wizards an der Elder-Conference teil. Ich vermute, daß dies bisher geduldet wurde, da es nicht so viele offizielle Teilnehmer gab, wie noch vor einigen Monaten. Außerdem trauen sich nur wenige Wizards, die Elder-Conference zu besuchen oder sich dort zu Wort zu melden.

Die Bedeutung der Elder-Conference wird unterschiedlich beurteilt, je nachdem, wen man fragt. Spricht man mit einem der Archwizards oder dem Gott, so erfährt man, daß die Entscheidungen selbstverständlich von der Administration getroffen wird und die Elder-Conference dabei lediglich als beratendes Gremium fungiert. Unterhält man sich jedoch mit einigen Elder Wizards, so hört man, daß die Entscheidungen von den Teilnehmern der Elder-Conference getroffen werden, womit diese die Stelle eines — allerdings benannten, nicht gewählten — Parlaments innehätte. Die Entscheidungsstrukturen sind an dieser Stelle nicht eindeutig geklärt. Vermutlich ist, wie so oft, alles ein bißchen der Fall.

Von der Stufe eines Elder Wizards an haben die Spielenden die Möglichkeit, auf Unix-Ebene und innerhalb des MUDs nicht nur lesend, sondern auch schreibend auf die Dateien der Mudlib zuzugreifen. Dadurch können zum Beispiel Fehler, die in fremden Objekten auftreten und einem Spieler Schwierigkeiten machen, von Elder- oder Archwizards behoben werden, auch wenn der betreffende Wizard nicht im Mud anwesend ist. Dieses Schreibrecht birgt allerdings auch die Möglichkeit, ungebeten oder gar mißbräuchlich auf fremde Dateien zuzugreifen, was gelegentlich leider vorkommt und dann Anlaß zu Konflikten gibt. Obwohl die Position eines Elder Wizards laut Definition nur Rechte und keine Pflichten mit sich bringt, bekommen Elder Wizards doch immer wieder Aufgaben von den Archwizards oder dem Gott übertragen. Zum Beispiel werden alle Objekte, die Wizards programmieren, geprüft, bevor sie in den Teil des MUDs eingebunden werden, der Spielern zugänglich ist. Es wird zum einen überprüft, ob neue Objekte in die Fantasy-Umgebung des MUDs passen, ob sie sorgfältig ausgearbeitet und interessant gestaltet sind, und zum anderen getestet, ob sie technisch fehlerfrei sind. Diese Prüfung soll normalerweise von den Archwizards vorgenommen werden, die angesichts ihrer geringen Zahl mit dieser Aufgabe (neben ihren anderen Pflichten im MUD) jedoch überfordert wären. Daher wird diese Aufgabe zuweilen an Elder Wizards delegiert. Indem die Elder Wizards diese und andere wichtige Aufgaben, die sie quasi zur Verwaltung des MUDs werden lassen, übernehmen, geht ihr Einfluß über den, den ein Ehrentitel bietet, hinaus und sie werden qua positionem eine Art Vorgesetzte anderer Wizards.

Die Archwizards werden von der Administration aus den Reihen der Wizards und Elder Wizards ernannt, bisher ausschließlich aus der Gruppe der Elder Wizards. Archwizards haben im Gegensatz zu den anderen Gruppen tatsächliche Pflichten. Sie bilden zusammen mit einem (im Falle von TubMud) oder mehreren Göttern die Administration, die eigentliche Leitung des MUDs. Archwizards und Götter sind tatsächliche Vorgesetzte aller anderen Gruppen. Sie treffen alle wichtigen Entscheidungen und stellen gleichzeitig die Gerichtsbarkeit des MUDs dar. Konflikte, die nicht direkt zwischen den Beteiligten

⁶Es wird jedoch von jeder Elder-Conference eine wortgetreue Mitschrift für alle Wizards im MUD bereitgestellt.

gelöst werden, sowie Verstöße gegen die Wizards' oder Players' rules⁷ werden der Administration mitgeteilt, die dann mit den Beteiligten oder Beschuldigten im MUD spricht und gegebenenfalls eine Entscheidung fällt oder sogar Strafen verhängt. Ich werde in Teil 5.3 noch näher auf die Gesetzgebung und Gerichtsbarkeit in MUDS eingehen.

Die Administration des MUDs bildet seine Außenvertretung in der realen Welt. Sie führt Gespräche mit den Netzbetreibern und pflegt die Hardware, also den Rechner, auf dem das MUD vor Ort läuft. Aufgabe der Archwizards und des Gottes ist außerdem nicht nur die Leitung der MUD-Gesellschaft, sondern auch die technische Pflege von Mudlib und Gamedriver. Daher sind Archwizards im allgemeinen versierte Programmierer, die für alle Designentscheidungen der Mudlib und meist auch ihre Realisierung verantwortlich sind. Die Administration eines MUDs steht in den allermeisten Fällen als geschlossene Gruppe da und verfügt durch die Gewalt über Datenbasis, Hardware und die Position in der Hierarchie über absolute Macht.

Die Götter eines MUDs wechseln nur sehr selten, daher liegen darüber, wie der Gott oder die Göttin bestimmt werden, keine Erfahrungswerte vor. Ein neues MUD wird meist von einer Gruppe untereinander bekannter Personen eröffnet, die dann bereits miteinander abgeklärt haben, wer welche Aufgabe übernimmt. Ein MUD gehört quasi seinem Gott und die meisten MUDs existieren noch nicht lange genug, um einen Wechsel nötig zu machen. Es ist aber anzunehmen, daß die Mitglieder der Administration eine gemeinsame Entscheidung darüber treffen, wer von ihnen die Position des Gottes übernimmt. Nur, wenn sich keiner in der Lage fühlt, diese Aufgabe zu erfüllen, keiner sie übernehmen will oder keine Einigung möglich ist, wird man wohl auf Personen aus anderen Gruppen innerhalb der Wizards zurückgreifen.

5.1.2 Die Gruppe der Peers

Wie man Abbildung 5.1 entnehmen kann, bilden die Clans eine besondere Gruppierung innerhalb der Spieler, als Alternative für diejenigen, die nicht zum Wizard aufsteigen wollen. In den Clans können sich die Spieler zu Gruppen Gleichgesinnter zusammentun und sich gegenüber den anderen Spielern etwas individualisieren. Mit der Einführung der Clans sollte das Rollenspielelement im MUD etwas mehr hervorgehoben werden. Es gibt in TubMud mehrere miteinander konkurrierende Clans, also organisierte Gruppen von Spielern, die Spieler ab der 10. Erfahrungsstufe als Mitglieder aufnehmen dürfen. Die Clans sind je nach Vorstellung der Clanführer, der **Dukes**, mehr oder weniger demokratisch organisiert und unterscheiden sich dementsprechend in ihren Hierarchiestrukturen. Beispielsweise könnte ein Clan anstelle des Dukes einen gewählten Rat haben oder generell auf eine Hierarchie innerhalb der Mitglieder verzichten. Spieler können innerhalb der Clans im Rang aufsteigen und **Peer** mit unterschiedlicher Position⁸ werden. Dabei werden sie entweder vom Leiter des Clans auf ihre Position erhoben oder gewählt, wobei es nach Aussage von Peers durchaus zur Änderung unerwünschter Wahlergebnisse kommen kann. Ich werde im folgenden auf diese „ungewöhnliche“ Interpretation von Demokratie noch eingehen.

⁷siehe Anhang

⁸Die Positionen können innerhalb der Clans nach Belieben festgelegt werden.

Die Einführung der Clans hat zu bedeutenden Veränderungen in der gesellschaftlichen Struktur von TubMud geführt. Vorher sind fast alle Spieler nach dem Erlangen der nötigen Erfahrungspunkte Wizard geworden. Diejenigen, die keinen Spaß am Programmieren hatten, haben das MUD entweder nur noch als Begegnungstätte genutzt oder haben es nach einiger Zeit ganz verlassen. Neben der Gruppe der Wizards ist eine weitere Gruppe herangewachsen, die ebenfalls Einfluß auf die Entwicklung von TubMud nehmen will. Die Peers sehen das MUD nicht so sehr aus der Sicht derjenigen, die es als Ganzes am Laufen halten wollen, sondern mehr aus der Sicht der Spieler, die nicht hinter die „Kulissen“ schauen können und neue gesellschaftliche Strukturen innerhalb der Spieler schaffen wollen.

Seit der Einführung der Clans im Herbst 1993 ist die Zahl der neuen Wizards stark zurückgegangen. Viele, die keine Programmiererfahrung haben, bleiben lieber Spieler und treten in einen der Clans ein, anstatt Wizard zu werden und sich mit dem Programmieren von MUD-Objekten auseinanderzusetzen. Manche probieren es auch aus und lassen sich dann nach kurzer Zeit zum Spieler zurückstufen. Es gibt durch diese Entwicklung jetzt zwar weniger Spielende, die sich ganz aus dem MUD zurückziehen, aber ein Mangel an Programmierernachwuchs macht sich bereits bemerkbar. Wo es lange einfach war, auch für größere Projekte genügend Interessierte zu finden, ist es jetzt schwierig geworden, manches nur am Laufen zu halten. In den letzten Wochen bis zur Fertigstellung dieser Arbeit konnte ich verstärkt bemerken, daß ehemalige Clan-Mitglieder Wizard werden, in der Absicht, speziell für die Clans zu programmieren. Es ist momentan noch nicht klar, wie sich dieser Trend weiter entwickeln wird. Im Extremfall besteht natürlich die Möglichkeit, daß ein MUD untergeht, wenn sich niemand mehr findet, der die Software pflegt. Dies ist aber für TubMud in absehbarer Zeit noch nicht zu erwarten.

Die strenge Hierarchie in beiden Gruppen wird von den Spielenden völlig unterschiedlich aufgenommen. Manchen scheint sie eine Sicherheit zu geben, die sie im realen Leben vermissen, andere wünschen sich mehr Demokratie und versuchen, die Strukturen zu verändern. Allen, die sich häufig in MUDs aufhalten, ist jedoch gemeinsam, daß sie die Gesellschaft innerhalb des MUDs, ihre Mitglieder und damit natürlich auch entstehende Diskussionen oder Konflikte sehr ernst nehmen. Erlebnisse bleiben nicht innerhalb des MUDs, sondern werden von den Spielenden mit in die reale Welt genommen. Sie wirken sich somit auf das Verhalten der Spielenden in der realen Welt ebenso wie in der virtuellen aus.

5.2 Was machen Leute im MUD?

Die Frage, was man in einer virtuellen Welt macht, habe ich bereits im zweiten Kapitel angerissen und möchte hier näher darauf eingehen. Natürlich beschäftigen sich viele Spieler damit, das MUD zu erkunden, Rätsel zu lösen, Monster zu überwinden, ihre gewählte Rolle auszuschnücken und zu spielen, oder als Wizard neue Objekte zu programmieren. Aber gleichzeitig dient das MUD auch als Kommunikationsort, ähnlich einem Dorfplatz, der allerdings virtuell und weltumspannend ist. An vielen „schwarzen Brettern“ kann man aktuelle Diskussionen verfolgen und seine eigene Meinung kundtun. Für Gespräche bieten sich eine Vielzahl geeigneter Orte an, wie das Gildenhause, das jedem Spieler bekannt ist,

eine der zahlreichen Kneipen oder einfach der Kirchplatz. Wie Curtis⁹ in LambdaMOO beobachtete, spielt Kommunikation für die Besucher virtueller Welten eine große Rolle. Die Kommunikation zwischen den Spielern und der gemeinsame Wunsch, die virtuelle Welt wie eine tatsächliche zu behandeln, bilden die Basis dafür, daß sich in MUDs eine Gesellschaft bildet¹⁰. Die Gemeinschaft im MUD ist der Grund dafür, daß MUDs auch nicht einfach nur Spiele sind. Die MUD-Gesellschaft ist so lebendig wie jede reale Gemeinschaft und unterliegt ähnlichen Schwankungen, Trends und Entwicklungen. Je länger jemand sich in der virtuellen Welt aufhält, desto höher ist der Grad der Identifizierung mit der virtuellen Person, durch die in dieser Welt gehandelt wird. Als Folge wird eine im Mud geäußerte Bemerkung oder ein im MUD erfolgter Eingriff in die Persönlichkeitsrechte von vielen Spielenden ähnlich schmerzlich empfunden, wie es in der realen Welt der Fall wäre.

5.2.1 Kommunikation ohne Stimme

Die Gesellschaft im MUD unterscheidet sich von realen Gesellschaften und ihre Mitglieder gehen anders miteinander um, als es in der realen Welt möglich ist. Jede Kommunikation findet über Text statt und auch Gefühle werden textuell dargestellt. Da kaum jemand so schnell auf der Tastatur tippen kann, wie er oder sie spricht, werden in MUD-Konversationen viele Abkürzungen benutzt und Sachverhalte knapp bis hin zur Mehrdeutigkeit dargestellt¹¹. Wichtiges Element sind zum Beispiel mit ASCII-Zeichen dargestellte „Smilies“, mit denen ein ironischer Unterton oder ähnliches angedeutet werden soll¹². Man kann sagen, daß MUD-Kommunikation auf diese Weise verkürzt abläuft. Gleichzeitig ermöglichen es die unterschiedlichen Kommunikationswege, also „say-Kommunikation“ im Raum und „tell-Kommunikation“ über beliebige Entfernung innerhalb des MUDs hinweg, mehrere Gespräche oder unterschiedliche Gesprächsebenen zur selben Zeit zu führen. Häufig kommt es vor, daß man sich mit Personen im Raum unterhält und dann per tell von jemand anderem angesprochen wird. Da es jedoch wie in der realen Welt als unhöflich gilt, dem Gesprächspartner nicht die volle Aufmerksamkeit zu widmen, werden tell-Gespräche im allgemeinen stillschweigend weitergeführt. Finden zuviele unterschiedliche Gespräche zur gleichen Zeit statt, kann man dies manchmal an den scherzhaft tell-lag genannten langen Antwortzeiten des Gesprächspartners merken. Zuweilen passiert es, daß sich neben dem say-Gespräch im Raum ein tell-Gespräch mit ein- und derselben Person ergibt. Dieses tell-Gespräch, das ja im realen Leben so nie möglich wäre, stellt häufig eine Art Meta-Gespräch dar, in dem das offen „hörbare“ kommentiert wird oder zum

⁹The majority of players spend the majority of their active time on MUDs in conversation with other players. The mechanisms by which those conversations get started generally mirror those that operate in real life, though sometimes in interesting ways., [4], Seite 11/21

¹⁰Moreover, the MUD community depends on a richness of communication and the creation of social context. [...] Communication is necessary to the existence of the MUD and successful MUDs are likely to see a great deal of communication between players, which can then form a basis for familiarity and intimacy., Reid [3], Seite 36

¹¹hierzu: Curtis[4], Seite 13/21 und 14/21

¹²Beispielsweise ist :-) ein freundliches lächelndes Gesicht, mit dem eine kritisierende Äußerung entschärft werden könnte und ;-) ein zwinkerndes Grinsen, das einen ironischen Unterton andeuten kann. Weitere Anregungen sind in [31] zu finden.

Beispiel darüber gesprochen wird, wie eigenartig die Konversation gerade läuft oder wie ungewöhnlich das Gesprächsthema ist. In der Elder-Conference kann es passieren, daß über tell-Gespräche Kommentare ausgetauscht und Absprachen getroffen werden, die die Entscheidungen behandelte Themen beeinflussen. MUD-Gespräche sind also nicht **nur** eine Verkürzung der Kommunikation, sondern stellen unter Umständen auch eine Bereicherung der Gesprächsmöglichkeiten dar.

Genauso wie im realen Leben wird aber in Gesprächen und Diskussionen nicht nur gesprochen, sondern es findet nebenher auch eine Menge an Gestik, Mimik und anderen Reaktionen statt. Da Personen im MUD ja ausschließlich über ihre virtuellen Spieler agieren, fallen Reaktionen wie zum Beispiel spontane Mimik weg. Jedes Gefühl, jedes Lachen auf einen Scherz wird als Befehl über die Tastatur eingegeben und ein im MUD geäußertes Kichern heißt noch lange nicht, daß sich das jeweilige Gegenüber an seiner Tastatur, wo immer diese auch stehen mag, tatsächlich amüsiert. Dies bedeutet, daß Gefühle in den allermeisten Fällen kontrolliert geäußert werden und kaum jemand durch eine unbeabsichtigte Reaktion mehr über sich verrät, als er oder sie möchte.

Die Verkürzung der Kommunikation an sich ist meiner Meinung nach auch der Grund, warum Konflikte im MUD kürzer, aber gleichzeitig heftiger ablaufen als im realen Leben. Ich habe mehrfach beobachtet und erfahren „können“, daß Streitpunkte zum Beispiel an den schwarzen Brettern heftig diskutiert werden, es unter Umständen zu einer schnellen Eskalation, einem sogenannten *flame-war* kommt, der aber auch ebensoschnell wieder vorbei ist. Ich kann meine Beobachtung nur so beschreiben, als daß demnach Konflikte in MUDs fokussiert ablaufen.

Ich halte es für fraglich, daß man jemand über das Medium MUD überhaupt kennenlernen kann. Kann man denn, wenn das Gegenüber immer die Möglichkeit hat, sich einem unerwünschten Gespräch mittels Abbruch der Verbindung zu entziehen, oder wenn Gefühle immer nur absichtlich geäußert werden, überhaupt wissen, wer jemand ist?

Den meisten MUDdern bedeuten ihre sozialen Beziehungen innerhalb des MUDs, die dort gefundenen Bekannten und Freunde sehr viel¹³, sie nehmen sie genauso ernst wie Freundschaften im realen Leben. Auch ich kann aus meiner persönlichen Erfahrung sagen, daß ich gute Freunde durch das MUD gefunden habe. Ich denke, wenn man mit jemandem viele lange Gespräche in der virtuellen Welt geführt hat, bekommt man durchaus ein Bild von der Persönlichkeit seines Gegenüber. Trifft man den- oder diejenige dann in der Realität, dann ist man zuerst einmal meist überrascht. Es gibt eben im realen Leben verhältnismäßig wenig große Drachen, mächtige Zauberer oder schwarze Ritter und die meisten düsteren Helden sehen in Wirklichkeit ganz harmlos aus. Unterhält man sich dann wiederum eine Weile, dann erkennt man in den Äußerungen und selbst ein bißchen an den bevorzugten Bewegungen die aus dem MUD bekannte Person. Auf diese Weise habe ich einige Freundschaften im realen Leben geschlossen, die ihren Ursprung in der virtuellen Welt hatten.

5.2.2 Ich möchte ein Eisbär sein . . .

MUD-Gesellschaften entwickeln eigene Verhaltenskodizes. Im realen Leben wird man sich, wenn man andere trifft, kaum verbeugen oder knicksen. In TubMud ist dies eine

¹³siehe Bruckman[1], Kapitel 3.2

normale höfliche Begrüßung. Wodurch kommt es zu so einer heutzutage ausgefallenen Form der Höflichkeit?

Zum einen ist die Verbeugung eine standardmäßig vorgesehene Form der Gefühlsäußerung. Irgendjemand fand es irgendwann einmal angebracht, sich verbeugen zu können. Erschwerend kommt dann hinzu, daß es ja in der textuellen Welt des MUD so einfach nicht möglich ist, einen Raum zu betreten, kurz mit jedem Anwesenden (von denen man nicht unbedingt jeden bereits kennt) Augenkontakt zu haben und zu nicken, ohne dabei ein laufendes Gespräch zu unterbrechen. Da jede Gefühlsäußerung explizit ist, wurde offenbar eine Form gesucht, die kurz und nicht störend ist. Auch hier kommt wieder die Verkürzung der Kommunikation zum Tragen, indem mit der Verbeugung eine Bewegung gewählt wurde, die in der Vorstellung keine direkte Berührung zur Folge hat. Spielende gehen mit ihrem virtuellen Körper meist sehr bewußt um, und daher muß auf eine explizit von jemand anderer ausgeübte Berührung reagiert werden. Eine einfache Verbeugung in den Raum hinein erfordert jedoch keine Reaktion mit Gegenberührung oder Worten. Dies mag im ersten Augenblick so klingen, als herrschten in virtuellen Welten strenge Verhaltensregeln. Aber das genaue Gegenteil ist der Fall. Der Umgang miteinander ist durch die Anonymität so zwanglos, daß viele Spielende mehr aus sich herausgehen können, als sie es im realen Leben vermögen.

Das anfängliche Fehlen normierter Verhaltensweisen hat in den frühen Zeiten der MUDs dazu geführt, daß sehr zwanglos und fast freizügig miteinander umgegangen wurde, so daß sich vermutlich manche Spielende dadurch verletzt fühlten. Dies hat sich im Lauf der Zeit durch zahlreiche Diskussionen eingependelt. Es möchte zum Beispiel nicht jeder mit einem Klaps auf den Hintern begrüßt werden. Im realen Leben ist es undenkbar, daß man so etwas tut, denn man hat die Grundregeln des Sozialverhaltens bereits als Kind gelernt (hoffentlich!). In den MUDs existierten diese Regeln anfänglich nicht. Die Spielenden haben zwar ihre Erfahrungen aus dem realen Leben mit ins MUD gebracht, aber trotzdem mußten Regeln des Umgang miteinander erst individuell für diese neuen Welten entwickelt werden¹⁴. Im MUD umarmt, tritt, knuddelt und zerrauft man sich gegenseitig die Haare viel häufiger als in der Realität. Obwohl das etwas grob wirkt, sind doch die Nuancen von größter Bedeutung. Zwischen einem „liebvollen Knuddeln“ und einem „herzlichen Umarmen“ liegen Welten, die die Adverbien, die mit den Verben zum Ausdrücken von Gefühlen kombiniert werden, ausmachen und die von den Spielenden genauestens registriert werden.

MUDs bieten nicht nur einen freieren Umgang miteinander, sondern auch die Möglichkeit, sich von den eigenen Unzulänglichkeiten zu befreien. Einige Spieler wählen die Beschreibung ihrer virtuellen Person ähnlich ihrem wirklichen Aussehen, die meisten jedoch suchen sich ein völlig anderes Äußeres. Dies kann zwischen dem einer schleimigen Plasmawolke oder einem Keks und dem eines mächtigen Drachen oder einer zierlichen Fee liegen. Häufig drücken die Spielenden auf diese Weise aus, wie sie sich selbst wirklich sehen oder wie sie gern sein möchten. Es gibt auch keinen Grund, sich in einer virtuellen Welt nicht von den Fallstricken der realen Welt zu befreien. Zudem führt dieses „Wählen des eigenen Ich“ zu interessanten positiven Effekten. Beispielsweise kann jemand, der sich durch eine Narbe oder dergleichen entstellt fühlt und in der realen Welt mit kaum jemandem

¹⁴hierzu: Reid [3], Seite 88

spricht, seine Ängste vor Ablehnung abstreifen und unter Umständen durch in der virtuellen Welt geschlossene Bekanntschaften das Selbstvertrauen gewinnen, sich anderen mitzuteilen. MUDs sind Spielwelten, die einerseits die Sicherheit einer allgemein akzeptierten Anonymität bieten und andererseits eine „warme soziale Umgebung“¹⁵ sind, in der Spielende es wagen können, sowohl für physische als auch psychisch begründete Probleme Lösungswege zu suchen und auszuprobieren.

Amy Bruckman berichtet von ihr bekannten Spielern, die sich und ihre Probleme mit Hilfe ihres Spielers darstellen¹⁶ und anderen, die für unterschiedliche Stimmungen unterschiedliche Spieler benutzen und auf diese Weise mehr über sich erfahren. Bruckman sieht MUDs als Freiraum für die Arbeit an Identitätskonzepten und ist der Meinung, daß man, wenn man sich in diesen virtuellen Welten engagiert, nicht umhin kommt, ein größeres Verständnis für soziale Phänomene zu entwickeln¹⁷.

5.2.3 Der Stoff aus dem die Träume sind

In einer Zeit, die von zunehmender Individualisierung und dem Aufbrechen sozialer Strukturen, seien es Familie, Freundes- oder Kollegenkreis, geprägt ist, scheint doch gleichzeitig Kommunikation für die Menschen immer wichtiger zu werden. Obwohl es den Anschein hat, als holten sich die Spielenden via Internet das „Gut“ Kommunikation als Konsumartikel, so spricht doch die Ernsthaftigkeit, mit der viele ihre sozialen Bindungen im MUD sehen und pflegen, dagegen¹⁸. Für manche scheinen die Freundschaften in der virtuellen Welt sogar die der realen zu ersetzen, und sie verbringen wöchentlich viele Stunden am Computer und im MUD. Das MUDden kann dann suchtähnliche Formen annehmen, weswegen es auch an vielen Universitäten verboten ist und auch in der Bundesrepublik einige MUDs geschlossen wurden¹⁹. Howard Rheingold bemerkt hierzu, daß die Art des Umgangs mit dem MUD näher betrachtet werden muß, denn oft ist zwischen virtuoser Benutzung eines Werkzeuges (in diesem Fall des sozialen Werkzeuges MUD) und suchthaftem Umgang nur schwer zu unterscheiden.²⁰ Wie auch schon Pavel Curtis bemerkte, ist es nicht die Sucht nach einem bestimmten Unterhaltungsspiel, sondern nach Kommunikation, die manche Menschen tage- und nächtelang in MUDs eingeloggt sein läßt²¹. Amy Bruckman beschreibt, wie sich aber auch in diesen Fällen häufig ein Reflektionsprozeß der

¹⁵hierzu: Bruckman [1], Seite 11 und Reid [3], Seite 35 ff.

¹⁶„Foo has chosen a character description that is similar to his real self. His character is an emotionless Vulcan. Foo tells me that he has trouble expressing emotion and tends to always try to work things out himself.“, [1], Seite 21

¹⁷„MUDs are a workshop for the concept of identity. Many players notice that they are somehow different on the net than off, and this leads them to reflect on who they are in real life., [...] „Paradoxically, that world is first and foremost social. One cannot fail to develop a greater understanding of social phenomena through living within it.“, [1], Seite 23

¹⁸siehe Bruckman [1], Seite 24 ff., Rheingold [2], Einleitung und Reid [3], Seite 36 ff.

¹⁹Als Beispiel sei hier die Schließung von TappMud 1993 angeführt, deren Begründung in den Anhängen zu finden ist.

²⁰siehe [2], Seite 193

²¹„It should not be supposed, that this kind of problem is now commonly-understand phenomenon of „computer-addiction; the fact that there is a computer involved here is more-or-less irrelevant. These people are not addicted to computers, but to *communication*; the global scope of Internet MUDs implies not only a great variety in potential conversants, but also 24-hour-access.“, [4], Seite 16/21

eigenen Situation einstellt, der sich positiv auf das reale Leben der Spielenden auswirken kann.²² Ich habe selbst beobachtet, wie Leute, die über keinen ausgeprägten Freundeskreis verfügten und zeitweilig bis zu 50 Stunden in der Woche MUDDend verbracht haben, über das MUD Freunde im realen Leben gefunden haben und dann in der Folge weniger Zeit im MUD verbracht haben. In diesen Fällen konnte ich beobachten, wie der Prozeß tatsächlich mit einer persönlichen Weiterentwicklung einherging.

Ein weiterer, wichtiger Aspekt dieser Freundschaften ist, daß ein Freund oder eine Freundin aus einem anderen Teil der Welt einem sehr gut dabei helfen kann, über den Tellerrand der eigenen kulturellen Umgebung zu sehen. Ich kann also Amy Bruckmans Sicht vom „identity workshop, nur voll und ganz unterstützen bzw. aus eigener Erfahrung belegen.

5.2.4 The bright side of life?

Wie ich bereits ausgeführt habe, gibt die Anonymität den Spielenden eine gewisse Sicherheit, daß ihr normales Leben von den Geschehnissen im MUD unberührt bleiben kann. Aber wie schon die Schleier der Salome führen auch im MUD scheinbare Geheimnisse zu einem Spannungsfeld, das zuweilen knisternde Erotik über das Internet zu senden scheint. Da nur 10 bis 15 Prozent der Spieler in TubMud überhaupt weiblichen Geschlechts sind und davon auch nur ein Bruchteil von Frauen gespielt wird, geben manche Spielende die Versuche nicht auf, herauszubekommen, ob ein weiblicher Spieler im MUD auch von einer Frau geführt wird. Zuweilen kommt es zu heftigen Flirts, die in vielen Fällen für den beteiligten männlichen Spielenden zu einer herben Enttäuschung werden. Legendär ist der von Elizabeth Reid²³ dokumentierte Fall, bei dem ein Mann und eine Frau Spieler mit jeweils vertauschtem Geschlecht hatten, im MUD erst Freundschaft schlossen, sich dann ineinander verliebten, dann jeweils das wahre Geschlecht beichten wollten, es erst nicht glaubten, sich dann trafen und einige Zeit später heirateten. Obwohl dieser Fall glücklich ausging, so muß ich doch feststellen, daß das MUD als Eheanbahnungsinstitut hochgradig ungeeignet ist, denn nur in den seltensten Fällen entwickelte sich nach meiner Beobachtung aus einem heißen MUD-Flirt eine tatsächlich haltende Beziehung.

Das Tauschen der Geschlechterrollen ist eine in MUDs häufig praktizierte Spielvariante. Meist sind es Männer, die die Rolle einer Spielerin annehmen, und herausbekommen wollen, ob sie als Mann erkannt werden (bzw. ob sie andere Männer erfolgreich bezüglich ihres realen Geschlechtes täuschen können) oder wie Männer auf Frauen im allgemeinen reagieren. Oft ist die Überraschung groß, daß auf der einen Seite zwar Spielerinnen viel geholfen wird, aber als Gegenleistung auch oft Zärtlichkeiten im MUD erwartet werden. Mir ist im Lauf der Zeit aufgefallen, daß Männer, die die Rolle einer Frau annehmen, häufig viel aggressiver flirten, als die meisten Frauen es aus ihrem Rollenverständnis heraus tun würden. Offenbar entspricht dieses sexuell „kooperative“ Verhalten so sehr der Erwartungshaltung vieler Männer, daß sie immer wieder auf die scheinbaren Frauen hereinfallen.

Neben den sozialen Interaktionen wie Gesprächen und Flirts, gibt es in den meisten MUDs sogar die Möglichkeit, andere Spieler (unabhängig vom realen oder MUD-Geschlecht) zu

²²siehe Bruckman [1], Seite 32 ff.

²³siehe Appendix zwei [3]

heiraten und einige MUDs sehen sogar Programme vor, die sich wie kleine Kinder verhalten. Diese Hochzeiten sind meist große Ereignisse innerhalb des MUDs und werden teilweise wochenlang vorher vorbereitet. In den meisten Fällen fühlen sich die Spielenden einander tatsächlich verbunden, aber nur selten sind diese „MUD-Ehen“ meiner Beobachtung nach von langer Dauer.

Auch im MUD wird das **Nein** einer Spielerin nicht immer auch als Ablehnung gewertet und es kommt leider immer wieder zu Übergriffen. Die Grenzen des Flirtens werden zuweilen von männlichen Spielern ignoriert und da mit den **Emotes**²⁴ so gut wie jeder Satz gebildet werden kann, der mit dem eigenen Namen beginnt, sind der Phantasie leider (in diesem Falle) keine Grenzen gesetzt. Hier kommt auch wieder zum Tragen, daß das MUD eben nicht „nur ein Spiel“ ist. Zwar kann sich eine ‚überfallene‘ Spielerin jederzeit aus dem Spiel ausloggen, aber zum einen hieße dies „das Feld dem Angreifer“ zu überlassen und zum anderen schlägt die Identifikation mit der eigenen Spielfigur hier gewissermaßen grausam zu. Ich kann aus eigener Erfahrung berichten, daß auch, wenn nur die Spielfigur der virtuellen Welt Mißbrauch ausgesetzt war, das elende Gefühl und die zitternden Knie ausgesprochen real gewesen sind. Im Gegensatz zum realen Leben, wo ich im Falle eines Überfalls immer noch versuchen kann zu kämpfen, habe ich diese Möglichkeit im MUD nicht. Ich kann zwar den Angreifer per Emote treten, aber erfahrungsgemäß ist dessen Identifikation mit der eigenen Spielfigur nicht groß genug, als daß er damit ernsthaft getroffen werden könnte. Hier ist es so, daß die Anonymität des MUDs einigen Spielenden offenbar die soziale Hemmschwelle vor Gewalt nimmt und sie im MUD ausprobieren, was sie in der realen Welt nicht tun können. Der Mißbrauch scheint den Tätern für das normale Leben keine offensichtlichen sozialen Folgen zu haben.

Nach einigen Gesprächen mit anderen Spielerinnen habe ich festgestellt, daß analog zum realen Leben jede dachte, es sei nur ihr passiert, und den Vorfall für sich behalten hat. Einige hatten zudem noch das Pech, auf verständnislose männliche Wizards zu treffen, als sie um Hilfe riefen, die der Meinung waren, im realen Leben „wär’s doch wohl schlimmer“ gewesen. Angesichts dieser Tatsachen plane ich zur Zeit in TubMud mit einigen Mitspielerinnen ein Projekt, daß diesem negativen Aspekt der virtuellen Welt etwas mehr Öffentlichkeit und den Spielerinnen etwas mehr Hilfe verschaffen soll.

Da auch männliche Spielende, die eine weibliche Rollen annehmen, ähnliche Erfahrungen in der Rolle als Frau machen, wirkt sich das MUD an dieser Stelle auf einige Männer in Bezug auf das Rollenverständnis positiv aus.

Der Gedanke der Täter, daß sie ja immer noch mit einem neuen Spieler von vorn anfangen könnten, wenn sie sich mit dem bisherigen unbeliebt gemacht haben, und daß die Anonymität auf dem Internet in dieser Hinsicht absolut ist, stimmt erfreulicherweise nicht. Es gibt in MUDs immer wieder Wizards, die sich verpflichtet fühlen, derartige Vorgänge im Auge zu behalten und einige bisher unbekannte Spieler, insbesondere nach einem Übergriff, unauffällig zu überprüfen. Zuweilen ergeben sich genügend Hinweise und einem erneuten Mißbrauch kann durch ein ernstes Gespräch, eine wohlgemeinte Drohung oder die Entfernung des Spielers aus der MUD-Welt vorgebeugt werden²⁵.

²⁴Emote ist der in der MUD-Welt allgemein gebräuchliche Begriff für die zum Ausdrücken von Gefühlen vorgesehene Befehle.

²⁵Siehe Playersrules eins und zwei in den Anhängen.

5.3 Manchmal sind Anarchie und Diktatur Schwestern

Amy Bruckman spricht von MUDs als ‚identity workshop‘. Elizabeth Reid geht über diese Definition hinaus und bezeichnet sie, indem sie JennyMUSH²⁶ als Beispiel anführt, als wertvolle soziale Werkzeuge. Howard Rheingold erhofft sich durch die virtuellen Welten eine Wiederbelebung der Demokratie durch die Bürger²⁷, warnt aber gleichzeitig auch eindringlich vor der zunehmenden Kommerzialisierung der weltweiten Netze und der damit einhergehenden Machtkonzentration.

Auch innerhalb der virtuellen Welten spielt die Gratwanderung zwischen demokratischem Umgang miteinander, strenger Hierarchie und Machtkonzentration eine große Rolle. Ich habe bereits umfangreich die Hierarchiestruktur von TubMud und die Zusammenhänge von Status und Rang eines Wizards dargestellt und möchte jetzt näher auf diesen Konflikt eingehen.

Einerseits geht es zwar wie schon beschrieben in MUDs sehr zwanglos zu. In TubMud darf jeder Wizard die Programme der anderen einsehen und der produzierte Code „gehört“ der MUD-Gemeinde gemeinschaftlich²⁸. Probleme der Gemeinschaft, seien sie technischer oder sozialer Natur, werden nach Möglichkeit unbürokratisch und durch Gespräche gelöst — solange keiner der Wizards höheren Ranges persönliche oder allgemeine Interessen gefährdet sieht und einschreitet. Wie kommt es zu diesen Strukturen, die gleichermaßen Elemente von Anarchie und Diktatur enthalten?

Die Menschen, die virtuelle Welten ins Leben rufen und sich am gesellschaftlichen Leben darin beteiligen, bringen natürlich ihre Vorstellungen von Gesellschaft aus dem realen Leben mit in die virtuelle Welt. Auf diese Weise entstehen Strukturen, die der Vorstellung der Beteiligten von Demokratie entsprechen, bzw. einem Kompromiß aus diesen Vorstellungen und dem, was die Beteiligten glauben, an Demokratie zulassen zu können, darstellen.

Als TubMud noch erheblich kleiner als jetzt war, lag die Leitung des MUDs auch schon klar bei der Administration. Es fanden unregelmäßig Wizard-Conferences statt, auf der alle interessierten Wizards mitdiskutieren konnten. Diese Konferenzen hatten die Funktion der heutigen Elder-Conferences, nur hatten alle Wizards das Recht, aktiv teilzunehmen. Die Administration hat laut Aussage von Demos, dem Gott von TubMud, auch damals die Diskussionen der Wizard-Conference nur als Entscheidungshilfe angesehen. Als das MUD mehr und mehr „Einwohner“ bekam, wurden die Wizard-Conferences zunehmend unübersichtlicher und schließlich zugunsten der Elder-Conferences abgeschafft. De facto haben Elder Wizards dadurch nicht mehr Rechte als „normale“ Wizards, sondern einfach mehr Macht. Da Elder Wizards jedoch benannt werden, und zwar in den meisten Fällen wegen ihrer technischen Qualifikation, kann von einer Demokratie bzw. der Elder Conference als Parlament eigentlich keine Rede sein. Trotzdem wird von technischen auf die sozialen Fähigkeiten geschlossen, indem Elder Wizards soziale Entscheidungen für die Gemeinschaft des MUDs treffen bzw. die Administration dahingehend beraten oder ganz

²⁶Ich werde im folgenden Abschnitt noch auf JennyMUSH eingehen.

²⁷hierzu: [2], Seite 26 ff.

²⁸Siehe auch Copyrightabsprache, genannt ‚Agreement‘, im Anhang.

einfach mit dem Gewicht ihrer Position Meinungen vertreten und in Konflikte eingreifen. Zum einen hat dies zur Folge, daß, zumindest nach meiner Beobachtung, das MUD von Spielern mit einer weitgehend technokratischen Weltsicht geleitet wird und nur in seltenen Fällen ausgesprochene Nicht-Technokraten in einflußreiche Positionen kommen. Zum anderen wird diese Herrschaftsform durch die technischen Strukturen gefestigt.

Technische und soziale Strukturen stehen im MUD in direktem Zusammenhang. Die Hierarchiestruktur ist in den Programmen des MUDs direkt festgelegt, indem zum Beispiel nur Wizards eines bestimmten (höheren) Ranges bestimmte Programme benutzen dürfen oder mittels Zugriffsrechten uneingeschränkter Zugriff auf die Dateien des MUDs haben. Diejenigen, die über die technischen Strukturen entscheiden, legen somit die sozialen Strukturen mit fest, da die Position in der Hierarchiestruktur über Mitspracherechte entscheidet und umgekehrt. Hinzu kommt, daß die meisten Spielenden sich bei einer Erhöhung des Ranges für das MUD meist mehr verantwortlich fühlen als vorher. Dies kann dazu führen, daß sie überempfindlich darauf reagieren, wenn sich ein anderer Spieler nicht so genau an die Regeln hält, oder daß sie versuchen, Mitsprachemöglichkeiten „niederer Ränge“ einzuschränken. Beispielsweise könnten die offiziellen Teilnehmer der Elder-Conference beschließen, „normale“ Wizards von der Konferenz wieder auszuschließen, da deren Anwesenheit ja nur geduldet war, oder unliebsame Diskussionen auf der Konferenz unterdrücken. Entscheidend ist wiederum das Demokratieverständnis der einzelnen Spielenden. Erfahrungsgemäß strebt jemand, der Hierarchie akzeptiert und für wichtig hält, mehr danach, in dieser Hierarchie aufzusteigen, als jemand, der sie als nicht relevant ansieht. Dies kann sich auf das Bild auswirken, das die „Hierarchie-Akzeptierenden“, den übergeordneten Entscheidungsträgern von sich vermitteln. Sind sie dann in der Hierarchie aufgestiegen, versuchen sie bewußt oder unbewußt, ihre Position abzusichern. Diesen Mechanismus der realen Welt kann man ebenso in der virtuellen beobachten.

Stellt man sich die Frage, warum für virtuelle Welten offenbar Hierarchiestrukturen gewählt wurden, die nicht freier sind als die der realen Welt, obwohl diese Welten ja viel kleiner und daher einfacher zu ändern sind, kann man eigentlich nur bei dem Bedürfnis der Menschen nach Ordnung und Sicherheit landen. Trotzdem kommt es auch hier zu Veränderungen, wie das folgende Beispiel deutlich macht. Analog zum großen „Lauschangriff“ existiert in TubMud das **snoopen**. Ursprünglich war es dazu gedacht, daß ein Wizard mit Hilfe der Spieler Fehler in eigenen Programmen finden konnte. „Snoopt“ ein Wizard einen Spieler, so hat er sozusagen eine Sonde in dessen Kopf und erhält alle Ausgaben des MUD-Programms, die auch der Spieler erhält, ohne daß der Spieler dies bemerkt. Gleichzeitig war das snoopen bis vor einiger Zeit in TubMud entsprechend den Hierarchiestrukturen geregelt. Das heißt konkret, daß Wizards Player snoopen konnten, ohne bemerkt zu werden, Elder Wizards ihrerseits normale Wizards und Player snoopen konnten usw. Wizards gleichen Ranges mußten sich gegenseitig gestatten, sich zu snoopen. Auf diese Weise konnte auch unbemerkt überprüft werden, ob ein Spieler sich durch einen Programmfehler einen unerlaubten Vorteil verschafft oder ein Wizard seine Möglichkeiten mißbräuchlich nutzt. Da bis vor einiger Zeit erst vom Rang eines Elder Wizards an geprüft werden konnte, wer wen snoopt, konnten Wizards vom Rang eines Elders oder höher von normalen Wizards nicht überprüft werden, aber ihrerseits jederzeit selbst unbemerkt

snoopen. Snoopen ist ein Thema, das laut Rheingold²⁹ die Spielenden in den virtuellen Welten in regelmäßigen Abständen wieder beschäftigt, da es immer wieder zu Mißbrauch kommt und zum Beispiel private Gespräche belauscht werden. Findet dieser Mißbrauch auch noch durch Wizards höheren Ranges statt, wird er zudem auch kaum bemerkt.

In TubMud kam es nach vielen langen Diskussionen und Streitigkeiten zu einer Regelung, die ich als Fortschritt in Hinblick auf Demokratie und Transparenz bezeichnen kann. Mittlerweile muß jedes snoopen gestattet oder eine Begründung angegeben werden, die in einer Datei dokumentiert wird. Außerdem kann nun zumindest jeder Wizard überprüfen, ob im MUD irgendjemand den Befehl snoop benutzt. Auf diese Weise kommt es zu einer Art sozialer Kontrolle, die aber zugegebenermaßen keine absolute Sicherheit bietet. Für Wizards, die ihrem Rang nach Schreibrecht auf die Mudlib haben, ist es noch immer rein technisch möglich, ungestattet und undokumentiert zu snoopen. Allerdings muß jedem, der auf diese Weise snoopt, deutlich klar sein, daß sein Tun nicht legal im Sinne des MUDs ist, und ein Verstecken hinter den scheinbaren Notwendigkeiten des eigenen Rangs ist nicht mehr so ohne weiteres möglich. Es bleibt an dieser Stelle zu sagen, daß es wohl doch keine technische Lösung für ein soziales Problem geben kann, aber immerhin ist es beachtlich, wie trotz der nicht demokratischen Struktur des MUDs Veränderungen durchgesetzt werden können.

Von TubMuds Gott, Demos, habe ich in einem Gespräch erfahren, daß er diese Regelung als teilweise Niederlage der von ihm verfolgten „Philosophie“ von TubMud sieht. Nach seinen Vorstellungen hätte es auch mit dem snoop-Befehl auf die Dauer dazu kommen sollen, daß jeder Einzelne eine besondere Verantwortung im Umgang mit diesem Befehl erlernt und ihn bewußt nicht einsetzt. Dies ist nach meinen Informationen bei keinem der Wizardränge eingetreten. Laut Demos ist die oben beschriebene technische Lösung des Problems nur umgesetzt worden, um wieder Ruhe in der Gesellschaft von TubMud zu schaffen. Er hält es für beachtlich, „wie eine Scheinlösung solch einen Eindruck hinterlassen kann“. Offenbar hat die von Demos erhoffte persönliche Entwicklung bei einigen Wizards nicht wie gewünscht eingesetzt. Vielleicht wäre es dazu notwendig gewesen, bereits viel früher klar zu machen, wie empfindlich der snoop-Befehl von vielen Spielenden beurteilt wird und daß eine Anwendung nur in seltenen Fällen angebracht ist.

Genau, wie das System der Hierarchie vielen Spielenden offenbar das Gefühl einer geordneten kontrollierbaren Welt vermittelt, wird die Notwendigkeit zu snoopen, um Fehlverhalten aufzudecken, prinzipiell von vielen Bewohnern von TubMud akzeptiert, unabhängig davon, ob diese Notwendigkeit nun real besteht oder nicht. Es wird gewünscht, daß zur Aufrechterhaltung von Ordnung und Sicherheit gesnoopt wird, solange keine privaten Gespräche belauscht werden. Das snoopen wird nicht grundsätzlich als Eingriff in die Persönlichkeitsrechte verstanden. Natürlich ist ein Mißbrauch des snoop-Befehls in keinsten Weise sicher zu verhindern, aber das Bedürfnis nach Sicherheit scheint, wie beim großen „Lauschangriff“ in der realen Welt, größer zu sein als die Angst vor Mißbrauch. Auch hier sieht man, daß sich gesellschaftliche Tendenzen der realen Welt im MUD widerspiegeln.

Ich denke, man muß sich darüber klar werden, daß, obwohl immer wieder versucht wird, demokratische Elemente in MUDs einzubringen, sie grundsätzlich nicht demokratisch an-

²⁹siehe [2], Seite 188

gelegt sind. Dies sieht man schon ganz einfach an der uneingeschränkten Macht der MUD-Götter, und es gilt für abenteuerorientierte MUDs in gleichem Maße wie für soziale. In den sozialen MUDs, die egalitärer wirken, sind lediglich die Machtmechanismen andere. Wie Elizabeth Reid beschreibt, ist in MUDs Macht buchstäblich die Macht, die Welt zu verändern.³⁰ Es ist die Macht über die Datenbasis genauso wie die Macht über das soziale Gefüge, die aus der Kontrolle über die Welt folgt. In sozialen MUDs liegt die Macht über die Datenbasis bei den Wizards und wird ausgeübt über Mechanismen wie die Zuteilung von Speicherplatz. Rangunterschiede gibt es auch dort, auch wenn sie nicht so klar erkennbar sind wie in den abenteuerorientierten MUDs. Auf der ‚World Mud Conference, habe ich mich im Juni 1994 in Linköping/Schweden mit den Göttern zweier schwedischer MUDs unterhalten, die in einem Workshop berichtet hatten, daß in ihren MUDs die Positionen der Elder Wizards (oder vergleichbare Ränge) durch Wahlen besetzt wurden. Zunächst war ich über diese Informationen sehr erfreut, bis ich durch Nachfragen erfuhr, daß die Wahlergebnisse im Falle unerwünschter Kandidaten im Zweifelsfalle durchaus auch einmal entsprechend abgeändert würden. In weiteren Unterhaltungen erfuhr ich, daß in den MUDs, in denen gewählt wird, eine Destabilisierung der Gesellschaft durch zum Beispiel radikale Kandidaten gefürchtet wird, die durch das Fälschen des Wahlergebnisses verhindert werden soll. Andere MUDs, wie zum Beispiel TubMud lehnen Wahlen laut Aussage des Gottes ab, weil mit den Wahlen auch unerwünschte Effekte, wie Stimmenkauf und Lobbyismus einhergehen würden.

Aus der Sicht einer MUD-Administration sind diese Befürchtungen durchaus verständlich, denn das soziale Gefüge eines MUDs ist zuweilen nicht so stabil, als daß es nicht auch untergehen könnte. Ich halte es allerdings für fraglich, ob eine Wahl wirklich den Untergang eines MUDs herbeiführen würde. In vier Jahren Studierendenvertretung habe ich gelernt, daß die Mühlen der demokratischen Gremienarbeit auch Quertreiber verändern können. Ich glaube, es wäre durchaus möglich, in MUDs etwas mehr Demokratie zu wagen, anstatt mit dem Wunsch nach Sicherheit eine gewisse Unmündigkeit zu akzeptieren. Vielleicht gehört es dann tatsächlich dazu, der Gefahr des Untergangs ins Auge zu sehen. Das bliebe abzuwarten. . .

Immer wieder kommt es in MUDs zu Mißbrauchssituationen, sei es das absichtliche Ausnutzen eines Programmfehlers, der Mißbrauch technischer Möglichkeiten durch Wizards oder gewalttätige Übergriffe. Verhandeln wie gegebenenfalls strafende Instanz sind in diesen Fällen die Wizards, wobei wie in Kapitel 2 beschrieben bei den Wizards der Rang nicht ohne Bedeutung ist.

Da ein Zugriff auf die reale Person des Spielenden in den meisten Fällen nicht möglich ist, werden Strafen bisher ausschließlich innerhalb des MUDs verhängt (also kein reales Gericht bemüht) und Konflikte ausschließlich innerhalb des MUDs geregelt. Meist kommt es zu einem klärenden Gespräch mit Beteiligung der Administration, das unter Umständen mit dem Verhängen einer Strafe endet. Eigentlich ist es innerhalb des MUDs nicht möglich, eine nachhaltige Bestrafung zu erzielen. Es gibt einfach nichts, das die reale Person eines zu strafenden Spielers treffen muß. Im schlimmsten Falle kann der Spieler aus dem Spiel entfernt und das Netz, aus dem er kam, für den Zugang nach TubMud gesperrt werden, falls er unter anderem Namen wieder auftaucht. Ein Wizard kann hingegen im

³⁰siehe [3], Seite 46

Rang erniedrigt (was höchst selten eintritt), oder ebenfalls aus dem Spiel entfernt werden. Handelt es sich zudem um einen Wizard von höherem Rang, so ist fraglich, ob die Verfehlung überhaupt öffentlich gemacht und zum Beispiel in der Elder Conference darüber gesprochen wird. De facto gibt es im MUD weder eine Art Untersuchungsausschuß, noch die Möglichkeit, auch nur so etwas wie ein antikes „Scherbengericht“ abzuhalten. Beides ist bisher nicht vorgesehen, die Macht der Wizards höherer Ränge ist absolut. Ich möchte gar nicht in Abrede stellen, daß sich die Administration eines MUDs große Mühe gibt, objektiv zu handeln, aber auch sie besteht nur aus Menschen, die für sich ja fehlbar sind und auch einmal mißbräuchlich handeln können. Aus diesen Gründen halte ich die Wirkung solcher MUD-Strafen unabhängig vom Spieler- oder Wizardrang für äußerst fragwürdig. Meiner Meinung nach muß die Wirkung von Konflikten innerhalb des MUDs nicht immer auf das MUD beschränkt bleiben. Beispielsweise habe ich selbst erlebt, wie mir jemand ein äußerst raffiniert programmiertes Objekt „angehext“ hat, das mich viermal stündlich mit einer zufällig aus einer umfangreichen Auswahl ermittelten Beleidigung versah, vergleichbar anonymen Briefen. In der realen Welt hätte ich den Täter, wenn er ermittelt worden wäre, nach allen Regeln der Kunst verklagen können. Im MUD ergeben sich gleich zwei Probleme: Wie weise ich nach, daß ein Wizard ein bestimmtes, im MUD befindliches Objekt programmiert hat? Und wie weise ich nach, daß der Wizard einer bestimmten realen Person zugehörig ist? Beide Fragen sind kaum zu beantworten, da es sich um ein elektronisches Medium handelt, bei dem die Beweislage in diesen Fällen noch ungeklärt ist. Trotzdem ist wohl, da immer mehr Menschen Zugang zum Internet haben und auch immer mehr Menschen ihre Zeit in virtuellen Welten verbringen, damit zu rechnen, daß derartige Fälle früher oder später die Gerichte beschäftigen werden.

5.4 Ein Ausblick

MUDs haben angefangen als Spiel, entwickelten sich zum sozialen Experiment und werden heute auch für „ernsthafte“ Zwecke benutzt. Mit Astro-VR haben Pavel Curtis und David A. Nichols ³¹ beispielsweise einen Begegnungsort für Astrophysiker geschaffen und arbeiten daran, auf der Basis dieses MUDs ein mit Videotechnik versehenes Konferenzsystem für wissenschaftliche Projekte zu installieren. Amy Bruckman hat für Medienwissenschaftler MediaMOO als gemeinsamen Arbeitsort initiiert und leitet dieses MUD seit dem Winter 1992/93 mit wachsendem Erfolg. Diese wissenschaftlich genutzten MUDs können sicherlich Konferenzen nicht völlig ersetzen, aber sie können in der täglichen Arbeit an internationalen Projekten eine deutliche Erleichterung der Arbeitsbedingungen sein. Wie viel angenehmer ist es, sich in einer virtuellen Welt zu einer Diskussion zu treffen, als elektronische Briefe auszutauschen oder zuweilen miteinander zu telefonieren. Der Kontakt zwischen den Projekt-Teilnehmern kann auf diese Weise intensiviert werden und Arbeitsergebnisse auf einfachem Wege ausgetauscht und besprochen werden.

Ein Projekt ganz besonderer Art ist JennyMUSH, ein virtuelles Hilfszentrum für Opfer von sexuellen Gewalttaten und Mißbrauch. JennyMUSH wird von einem Psychologiestudenten geleitet. Opfer von derartigen Gewalttaten können dort, wenn gewünscht, miteinander reden, aber auch auf die Hilfe von psychologisch geschulten Personen zurückgreifen. Viele

³¹nachzulesen in [5]

Opfer konnten dort in der Sicherheit der Anonymität und der scheinbaren Körperlosigkeit erstmals über ihre traumatischen Erlebnisse sprechen.³²

Mittlerweile sind MUDs nicht nur Gegenstand der Forschung und zur Forschung geworden, sondern es wird auch damit experimentiert, MUDs als Lehrmittel einzusetzen. Wie ich bereits im Kapitel 3 erwähnt habe, fanden im Wintersemester 1993/94 an der TU Berlin zwei studentische Seminare im Zusammenhang mit MUDs statt. Das eine als Basis gedachte Seminar diente der gemeinsamen Erarbeitung der Konzepte objektorientierter Programmierung und objektorientierten Designs. MUDs, mit ihrer Möglichkeit, Objekte anzufassen und ihre Interaktion gewissermaßen am eigenen Leib zu erfahren, sind ein besonders anschauliches Beispiel für objektorientierte Programmierung³³. Das Seminar nutzte ein MUD als Entwicklungsumgebung für objektorientierte Programmierung. Gleichzeitig konnten die Teilnehmenden sich ihre Arbeitsergebnisse im MUD gegenseitig vorführen und gegebenenfalls auch dort diskutieren. Das zur gleichen Zeit stattfindende zweite Seminar richtete sich an Studierende mit bereits vorhandenen Kenntnissen objektorientierter Programmierung. Ziel war die Entwicklung eines Game Drivers, der als verteiltes System auf einem Netzwerk laufen sollte³⁴. Nachdem in einem Semester die Konzepte für einen solchen verteilten Game Driver entwickelt wurden, ist die Gruppe nun mit der Implementierung beschäftigt.

³²Mitsamt einer denkwürdigen Anekdote nachzulesen bei Reid [3], Seite 40 ff.

³³hierzu: [24], Seite 6

³⁴siehe Abschlußbericht: [25]

„Im ‚Real life‘ gibt es keine Elder und Archwizards und selbst wenn, dann wäre niemand von uns einer von ihnen.“

HAL, Elder wizard in TubMud

Kapitel 6

Schlußbetrachtung

Weltweit ist in den letzten Jahren die Zahl der Menschen, die Zugang zum Internet, haben drastisch angestiegen. Während das Netz lange fast nur von Universitätsangehörigen bevorzugt aus technischen Bereichen benutzt wurde, sind es heute Menschen aus vielen unterschiedlichen Berufsbereichen, die sich das Netz teilen. Mittlerweile bieten sogar einige deutsche Zeitungen und Zeitschriften, wie zum Beispiel der Spiegel¹, Informationsseiten, Leserservice und Artikel über das Internet an. Der Bundestag hat zwar noch keine e-mail-Adresse, aber die schwedische oder amerikanische Regierung mag da als Vorbild dienen².

Es kann in meinen Augen kein Nachteil sein, wenn Bürger ihre Abgeordneten und Minister per e-mail erreichen, Fragen stellen und Informationsmaterial zu aktuellen Gesetzesentwürfen abrufen können. Mit dem Wunsch nach Information „rund um die Uhr“ werden die Computer, die diese Informationen frei Haus liefern, immer wichtiger. Die Menschen verbringen immer mehr auch Teile ihrer Freizeit „auf dem Netz“. Wie man an der steigenden Zahl von MUDs und ihrer „Bewohner“ sehen kann, entdecken sie auf ihren Reisen durch die internationalen Datennetze auch die virtuellen Welten. Gleichzeitig bemerkt man auch bei Gesprächen in den MUDs, daß mehr und mehr auch Mediziner, Juristen, Wirtschaftler sowie Menschen aus nicht-akademischen Berufen zur MUD-Gemeinschaft zählen. Ähnlich wie Howard Rheingold und Amy Bruckman habe ich die MUDs als virtuelle Gemeinschaften erlebt und beobachtet, wie einzelne Bewohner ihre Persönlichkeit mit Hilfe von Freunden in der virtuellen Welt erkannt und entwickelt haben.

MUDs werden aber nicht nur als Freizeitbeschäftigung, sondern auch als Versuchsfeld für soziologische Studien und als wissenschaftlicher Arbeitsraum genutzt, wie zum Beispiel MediaMOO und Astro-VR. JennyMUSH wird auch sicher nicht das einzige MUD bleiben, das dem Umgang mit spezifischen Problemen gewidmet ist und somit als soziales Werk-

¹Der Spiegel bietet Informationsseiten des World-Wide-Web sowie einen Leserbriefservice unter <http://spiegel.nda.net/hda/spiegel> an.

²Informationsseiten des schwedischen Parlamentes sind über World-Wide-Web unter <http://www.sb.gov.se/>, des US-Repräsentantenhauses unter <gopher://gopher.house.gov.1> und des US-Senates unter <gopher://gopher.senate.gov/1> abrufbar. Außerdem kann man den schwedischen Staatspräsidenten via e-mail unter statsministern@sb.gov.se und den US-Präsidenten unter president@whitehouse.gov anschreiben.

zeug dient, wie Elizabeth Reid es beschreibt³. Ich kann mich der Sicht von Rheingold, Bruckman und Reid⁴ zwar anschließen, glaube aber, es ist nötig noch weiter zu gehen und auch die Strukturen **innerhalb** dieser virtuellen Welten genauer zu untersuchen.

In den Zeiten der Informationsgesellschaft, in der wir heute immer mehr als einzelne Individuen leben, übernehmen Freundschaften in der virtuellen Welt teilweise die Funktionen der sozialen Gemeinschaft, der Familie oder des Freundeskreises, in die die Menschen früher eingebettet waren. Diese Gemeinschaften unterscheiden sich aber von den bisher üblichen. Die Freunde sind nicht wirklich da, es sind virtuelle Freunde, die einen zwar in der virtuellen Welt umarmen und trösten können, aber nur in den seltensten Fällen auch in der realen. Gleichzeitig kommen Menschen aus den unterschiedlichsten Teilen der Welt zusammen, manchmal ist ein Kennenlernen in der realen Welt gar nicht möglich. Soziale Beziehungen in der virtuellen Welt bleiben auch auf einem für die Beteiligten berechenbaren Maß. Geht jemandem ein Gespräch beispielsweise zu nahe, kann er sich immer, ohne daß der Gesprächspartner eingreifen kann, der Situation entziehen und sich ausloggen. Wenn diese sozialen Beziehungen einen Ersatz für reale darstellen, dann sind sie in meinen Augen unvollständig, weil sich die Personen immer hinter der Maske verstecken können, die ihnen die Spielerfigur bietet. Selbst, wenn man unwillkürlich zum Beispiel durch Wortwahl mehr über sich verrät, als einem bewußt ist, so zeigt man doch in den meisten Fällen in der virtuellen Welt nur das Gesicht, das man auch zeigen möchte.

Howard Rheingold sieht, wenn er von der virtuellen Gemeinschaft spricht, reale Menschen, die sich in der virtuellen Welt treffen, sich unterhalten, eventuell über Probleme reden und dort eine Gemeinschaft bilden. Er sagt zwar, daß seine virtuellen Gemeinschaften von ihm Besitz ergriffen und sein Begriff von Familie auf grundlegende Weise virtualisiert wurde, aber ich glaube, dieser Sozialesatinsprozeß geht sogar noch weiter. Indem Menschen einen Teil ihres sozialen Lebens in die virtuelle Welt verlegen, entsteht dort nicht nur, wie allgemein unumstritten ist, eine Gemeinschaft, sondern sogar eine Gesellschaft.

Die Gemeinschaften der virtuellen Welten verselbständigen sich. Sie bilden eigene Regeln des Zusammenlebens, des Umgangs miteinander, der Gesetzgebung (bzw. Regelbildung) und sogar eigene Formen von Justiz **innerhalb** der virtuellen Welt. Damit, daß sie auf dem Netz existieren, laufen die Programme selbst zwar auf einem Computer in einem bestimmten Land, der den dortigen Gesetzen unterliegt, aber die Welten liegen in einer Art Niemandsland. In den virtuellen Welten gelten die Gesetze, die sich die „Bewohner“ dort gegeben haben und die sind manchmal denen der realen Welt sehr ähnlich, manchmal unterscheiden sie sich aber auch davon. Einerseits ist es gut, daß die weltweiten Netze noch so wenig kontrolliert sind, denn dadurch ergibt sich der Freiraum, in dem die virtuellen Welten, die MUDs entstehen und wachsen. Andererseits droht, wie Howard Rheingold schon bemerkt, die Kommerzialisierung der MUDs analog zu Kommerzialisierung der internationalen Netze. Schon jetzt wird über Werbung auf dem Internet diskutiert, und viele Firmen stehen „in den Startlöchern“, das Internet mit Werbeseiten zu überziehen. Spielekonzerne warten darauf, daß graphische MUDs ausgereift sind, um sich ihren Marktanteil zu sichern und die Menschen mit noch mehr Unterhaltung anstelle von Information zu überschütten. Dies wäre dann vermutlich auch im Sinne einiger politischer Gruppen, denen es sicher ein Dorn im Auge wäre, wenn Bürger zu jeder Tages- und Nachtzeit die

³siehe Kapitel 5.3

⁴siehe Kapitel 5

Grundlagen politischer Entscheidungen überprüfen und so Transparenz der politischen Arbeit einfordern könnten.

Es ist zum einen zwar gut, daß das Internet zur Zeit noch so frei von Regelungen ist, unter anderem auch die Freiräume für virtuelle Welten einschränken könnten, aber andererseits müssen bei der zunehmenden Zahl von Teilnehmern Wege gefunden werden, die Interessen der einzelnen Netzbenutzer zu schützen. Nur wenn auf die Entstehung der Regeln von vielen einzelnen Benutzern Einfluß genommen wird, werden die Regeln auch die Interessen des Einzelnen widerspiegeln können. Ein Einschreiten gegen den Trend der Kommerzialisierung des Internets ist schwierig und könnte höchstens durch einen internationalen Konsens der Benutzer herbeigeführt werden.

Andererseits halte ich es aber auch für wichtig, die miniaturisierten Gesellschaften innerhalb der MUDs genauer zu untersuchen, da sie, wie bereits angeführt, von immer mehr Menschen professionell oder in der Freizeit genutzt werden. Wenn wir beobachten, welche gesellschaftlichen Strukturen sich die Menschen in diesen Welten schaffen, können wir Rückschlüsse auf die realen Gesellschaften ziehen. Meiner Meinung nach bauen sich die Menschen in den MUDs einerseits Welten so, wie sie sie haben wollen, aber sie leben dort auch immer wieder ihr Bedürfnis nach Herrschaft aus. Es werden positive wie negative Utopien gleichermaßen verwirklicht.

Ich glaube, daß in nicht allzu ferner Zeit nicht nur universitäre Projekte in MUDs diskutiert werden, sondern man sie auch für wirtschaftliche Konferenzen benutzen wird. Dabei entstehen Probleme, die ansatzweise schon heute durch Videokonferenz- und papierfreie Bürosysteme bekannt sind. Wenn ich mit einem am anderen Ende der Welt sitzenden Geschäftspartner einen Vertrag in einem MUD schließe, dann stellen sich für mich juristisch wichtige Fragen, wie:

- Wie stelle ich sicher, daß mein Gegenüber im MUD wirklich mein Geschäftspartner ist?
- Wie können Unterschriften in die virtuelle Welt eingebracht und überprüft werden?
- Welches Recht gilt für den gemeinsam geschlossenen Vertrag? Das aus dem Land eines der Geschäftspartners? Das des Landes, in dem das MUD auf einem Computer läuft?
- Kann es eine Bedrohung durch Wirtschaftsspionage via snoop-Befehl geben?

Aber nicht nur Geschäfte und wissenschaftliche Projekte werden über virtuelle Welten laufen. Es können auf dem Internet oder im MUD, wie bereits mehrfach beobachtet, Arbeitsstellen ausgeschrieben werden, und der Tag, wo die erste Beleidigung in einem MUD nicht mehr nur dort geregelt wird, sondern unsere Gerichte bemüht werden, ist sicherlich auch nicht mehr fern. Mit allen diesen Problemen müssen wir uns in der nächsten Zeit befassen, und für einige, zum Beispiel die rechtlichen, sind Lösungen noch nicht in Sicht.

Literaturverzeichnis

- [1] Identity Workshop: Emergent Social and Psychological Phenomena in Text-Based Virtual Reality, Amy Bruckman (asb@media-lab.media.mit.edu), 5.April 1992, erhältlich über ftp parcftp.xerox.com /pub/MOO/papers.
- [2] Virtuelle Gemeinschaft: Soziale Beziehungen im Zeitalter des Computers, Howard Rheingold, 1994, Addison-Wesley, Bonn
- [3] Cultural Formations in Text-Based Virtual Realities, Elizabeth Reid, 1994, erhältlich über ftp ftp.ee.mu.oz.au /papers/emr.
- [4] Mudding: Social Phenomena in Text-Based Virtual Realities, Pavel Curtis, 3.März 1992, Xerox Parc, erhältlich über ftp parcftp.xerox.com /pub/MOO/papers.
- [5] MUDs Grow Up: Social Virtual Reality in the Real World, Pavel Curtis und David A. Nichols, 19.Januar 1993, Xerox Parc, erhältlich über ftp parcftp.xerox.com /pub/MOO/papers.
- [6] Artikel ‚Sprechender Handschuh‘ aus der Zeitschrift Der Spiegel, Ausgabe Nr.23, 1992, Herausgeber Rudolf Augstein, Hamburg.
- [7] Artikel ‚Gesetzlos im Cyberspace‘ aus der Zeitschrift Der Spiegel, Ausgabe Nr.32 1994, Herausgeber Rudolf Augstein, Hamburg.
- [8] Artikel ‚Nadelstreifen im Cyberspace‘ aus der Zeitschrift Der Spiegel, Ausgabe Nr.11 1991, Herausgeber Rudolf Augstein, Hamburg.
- [9] Artikel ‚Schattengleich im Swimmingpool‘ aus der Zeitschrift Der Spiegel, Ausgabe Nr.34 1990, Herausgeber Rudolf Augstein, Hamburg.
- [10] Artikel ‚Engel zum Anfassen‘ aus der Zeitschrift Der Spiegel, Ausgabe Nr.32 1994, Herausgeber Rudolf Augstein, Hamburg.
- [11] Artikel ‚Zauberreiche aus Bits und Bytes — Wo digitale Götter walten‘ aus der Zeitung Die Zeit, Ausgabe 7. Oktober 1994, Herausgeber Helmut Schmidt.
- [12] Artikel ‚Künstliche Spiele — Praxis und Theorie interaktiver Installationen‘ aus der Zeitschrift c’t – magazin für computertechnik, Ausgabe Nr.12, 1993, Verlag Heinz Heise GmbH & Co KG.

- [13] Artikel ‚Daten-Ritter — Monsterjagd per Telefon: MUD‘ aus der Zeitschrift Power Play, Ausgabe Nr. 2, 1994, Markt & Technik Verlag AG.
- [14] Frequently asked Questions: Basic Information about MUDs and MUDDing, regelmässig veröffentlicht im USENET News-System von Jennifer Smith (jds@math.okstate.edu), Version vom 1.Mai 1994 (Archive: ftp.math.okstate.edu/pub/muds/misc/mud-faq).
- [15] Interactive Multi-User Computer Games, Richard Bartle (Richard%tharr.UUCP@ukc.ac.uk), Dezember 1990, erhältlich über ftp parcftp.xerox.com /pub/MOO/papers.
- [16] Advanced Dungeons & Dragons: Player’s Handbook, Second Edition, David Cook und andere, 1989, TSR Inc.
- [17] Zork and the Future of Computerized Fantasy Simulations, David Lebling, aus der Computerzeitschrift Byte, Ausgabe Dezember 1980, Seite 172-182, bearbeitet und im USENET veröffentlicht von Peter Barton (s892031@minos.xx.rmit.oz.au, pumpkin@phoenix.puz.uu.oz.au).
- [18] Have Computer, Won’t Travel, Virtual Meeting Place Opens at MIT Media Lab, Wade Roush, aus der Zeitschrift Technology Review Magazine, 1993, im USENET veröffentlicht vom Autor.
- [19] A History Of Mud, Alan Cox, 1993, im USENET veröffentlicht vom Autor.
- [20] Zen and the Art of the Internet — A Beginner’s Guide to the Internet, Brendan P. Kehoe, 1992, 2.Aufl., Prentice-Hall, Englewood Cliffs, NJ.
- [21] Programmieren in C, Brian W. Kernigham, Dennis M. Ritchie, 1990, 2. Aufl., Carl Hanser Verlag, München, Wien.
- [22] Künstliche Spiele, herausgegeben von Georg Hartwagner, Stefan Iglhaut und Florian Rötzer, 1993, Klaus Boer Verlag, München
- [21] Der Computer, eine Radikalisierung der Schrift?, Martin Fischer, 1994, erhältlich beim Autor, Berlin (email: fischli@cs.tu-berlin.de).
- [23] Das Simple Network Management Protocol (SNMP) — Darstellung und Bewertung des Quasi-Standards im Bereich Netzwerkmanagement, Diplomarbeit von Dietrich Kirsch, 1993, TU-Berlin.
- [24] Autonomes Seminar a_S OOPS!, Abschlußbericht und Auswertung, René Laufer, Irina Leyde, Klaus Rennecke, Jesper Richter-Reichhelm, Alexander Weidt, Jacob Wieland, 1994, erhältlich über ftp, morgen.cs.tu-berlin.de 7681 /ftp.
- [25] Autonomes Seminar a_S D.O.M.E, Alexander Weidt, 1994, erhältlich über ftp morgen.cs.tu-berlin.de 7681 /ftp.

- [26] LPmud, a programable multi user game, Lars Pensjö (lars@cs.chalmers.se), bearbeitete Version vom 3.7.93 von Amylaar, Ardanna, Demos, Macbeth, Marion, Foslay und anderen; erhältlich über [ftp ftp.cs.tu-berlin.de /pub/mud/lp](ftp://ftp.cs.tu-berlin.de/pub/mud/lp).
- [27] Electropolis — Communication and Community on Internet Relay Chat, Elizabeth Reid, 1991, erhältlich über [ftp ftp.ee.mu.oz.au /papers/emr](ftp://ftp.ee.mu.oz.au/papers/emr).
- [28] Object Oriented Programming, Brad J. Cox, 1986, Addison-Wesley, Bonn.
- [29] Introduction to Automata Theory — Language and Computation, John E. Hopcroft, 1979, Addison-Wesley, Bonn.
- [30] Big Dummy's Guide to the Internet — A round trip through Global Networks, Life in Cyberspace, and Everything, Adam Gaffin, Jörg Heitkötter, 1994, TeXinfo Edition 2.2, EFF — The Electronic Frontier Foundation.
- [31] Smileys, Compiled by W. Sanderson, Text by D. Dougherty, 1993, O'Reilly & Associates Inc.

Danksagung

Bei der Erstellung dieser Studienarbeit haben mir viele Menschen meiner Umgebung mit Rat und Tat zur Seite gestanden. Ihnen möchte ich an dieser Stelle danken:

Professor Dr. Bernd Lutterbeck, der es mir ermöglichte, eine Studienarbeit über ein noch sehr ungewöhnliches Thema wie virtuelle Welten zu schreiben.

Gerrit Oldenburg, der die Arbeit in weiten Teilen betreute, mir geduldig Hilfestellung, Anleitung und Kritik bot und auch, als ich furchtbar lange brauchte, nicht sauer war.

Anne-Katrin Schroeder-Lanz (Annabell@TubMud) und Martina Busch (Trillian@TubMud) für die Hilfe bei der Literatursuche.

René Laufer (Hal@TubMud), ohne den ich den Kampf gegen L^AT_EX wohl nie gewonnen hätte.

Alexander Weidt (Demos@TubMud), Klaus Rennecke (Myxectbo@TubMud) und Holger Parplies für ihre Hilfe insbesondere beim technischen Teil (und natürlich endloses Korrekturlesen). Zusätzlich Alexander Weidt (Demos@TubMud) für die vielen liebevollen „B.S.“ bei der Korrektur. :)

Jesper Richter-Reichhelm (Arkon@TubMud), Susanne Kloas (Myriel@TubMud) und Ingo Herpolsheimer für's n+1-te Korrekturlesen.

Meinen Nerven, dafür, daß sie wider mein Erwarten nicht aufgeben haben, den indischen Tee-Anbauern für die Unmengen Darjeeling, die ich während der Arbeit getrunken habe und meinen Katzen fürs stundenlange Schnurren neben meinem Arbeitsplatz, ohne das ich wohl nicht so lange am Schreibtisch ausgeharrt hätte.

Anhang A

Die Levelstruktur

Lvl	Experience	Male	Female	Creature
20	1500000	apprentice wizard	aapprentice wizard	apprentice wizard
19	1200000	grand master sorcerer	grand master sorceress	ferocious tyrannosaur
18	900000	master sorcerer	master sorceress	small tyrannosaur
17	600000	apprentice sorcerer	apprentice sorceress	vicious dragon
16	300000	warlock	witch	devious dragon
15	200000	enchanter	enchantress	small dragon
14	130000	magician	magicienne	powerful demon
13	90000	apprentice magician	apprentice magicienne	small demon
12	60000	conjurer	conjuress	beholder
11	40000	champion	deadly amazon	great monster
10	25000	warrior	amazon	experienced monster
9	17000	great adventurer	great adventuress	medium monster
8	11000	experienced adventurer	experienced adventuress	small monster
7	7500	small adventurer	small adventuress	threatening shadow
6	5000	experienced fighter	charming siren	shadow
5	3375	small fighter	siren	wraith
4	2250	master ranger	master ranger	bugbear
3	1500	lowrank ranger	lowrank ranger	furry creature
2	1000	simple wanderer	simple wanderer	simple creature
1	0	utter novice	utter novice	utter creature

Anhang B

Die Wizards' rules

These are the rules and code of conduct for a wizard.

1. You may not help, or meddle in the affairs of players, even if they have lost points due to a lost connection. They have to go to the post office and ask Demos, or use the bug command.
2. Never attack a player or kill him. If a player behaves badly, file the complaint to Demos.
3. Do not make 'deadly' traps. A player must never die because he didn't know what happens in the next room. If some room is very dangerous, make some hint (like the giant footprints beside the giant lair).
(addendum: I tend to make exceptions to this rule, please ask a lord if you have a question regarding to it.)
4. Never initialize the destination of an object outside your castle. If you make a monster that can walk out of the castle, be sure that it is a very nice monster.
5. Never generate messages that look like something they aren't (good and precise rule :-). That is, don't try to fool the player that someone says something when it isn't true etc.
6. If you have a 'test' character, make sure is isn't seen in the top score list. He must no be wizard either. He must also be registered as a 'test player' (see /room/office)
7. Try to avoid making devices, situations which make the players loose experience if they are adventuring or examining objects. Players should be rewarded for adventuring and discovering things. If they get killed...
Well, they will then loose some experience, but this is a special case!
8. The game is supposed to be in the 'long distant past', and thus no modern things should exist. If you want some kind of airplane, use a flying horse instead etc.

9. If you make available some kind of restaurant that sells food that heals, then the healing must cost on average 4 gp/hp (or more). The amount of healing must also be limited per reset. Generating healing items for free is only allowed if they heal at most 20 points, and are destructed. Not more than one such item may be generated per room and reset.
10. Teleporting items should be VERY restricted, and very rare. They must not allow teleportations to anywhere, but only to one or more predefined room. Never to a specific player. It must cost at least 50 spell points to use each time.
11. Weapons statistics are to be made with the following functions:
the value of a weapon with weapon class i is:
 $(i^4)/80$... or at least approximatley B-
and the weight of the weapon is:
 $(i + 3)/6$... or near (but NEVER have a negative weight...B-)
Naturally two-handed swords cannot allow simultaneous wielding and the wearing of shields, aswell as performing actions requiring two hands.
12. Armours are to follow the guidelines described in armour.list and armour.doc.
Naturally, shields must prevent the simultaneous wielding of two-handed weapons, and all other actions that require the use of two hands.
13. Magic items may only have one type of magic and a limited number of charges. They must not be easily accessible (being guarded by a monster does not suffice) and the computation of the effect of using a magical device should be based on the level of the caster (for example, an amulet of healing that heals five hp per level or a ring of teleport that works with probability $(5*\text{level})$ percent).
14. If an elder-/archwizard tells you to change a certain item, he does so for a reason. Implement the change immediately please.
15. PLEASE REMEMBER, ABOVE ALL ELSE, THAT THIS MUD STRIVES TO BE A *QUALITY* MUD, SO DON'T JUST BUILD VAST HACK-AND-SLAY ADVENTURES. HERE ARE A FEW CRITERIA TO CONSIDER WHEN IMPLEMENTING A QUEST:
 - (a) make it fun.
 - (b) make it realistic.
 - (c) have it *require* role playing and team activities.
 - (d) include some puzzle solving.
 - (e) DON'T IMPORT IT FROM ANOTHER MUD.

16. Do not make your objects publically accessible before an archwizard has approved it. This rule is to prevent half-finished items lying around in the MUD.

This means that you should change the DEST field of your castle.c to 'room/new_castles' and then write a note on the board in /room/new_castles with your reserved spot in /room for your castle and in the note as such, you could write your ideas on what you want your castle to be when it is ready...
happy hacking B-)

Anhang C

Die Players' rules

Part I

Welcome to TubMud.

First: This is a GAME. Players are here to have fun.

This is why you should treat everyone here as you would treat your friends in the REAL world.

Misunderstandings and hurt feelings also occur in electronic communication and can be as unpleasant to all sides as in REAL life. The MUD is not a place where you can do anything you can't do in REAL life. It is a place to have fun with other people. Politeness, friendliness and helpfulness are a guarantee that we all can have fun here.

When you get upset about something here, keep cool, ok? It is JUST a game and some problems are easy to fix. :)

Part II

Second: This is more than just a GAME. We play with each other and not against each other. We are all guests in this game and whoever makes life unenjoyable for the others risks removal from the game. (This is no empty threat. We do it rarely and we don't like to do it. But some people just don't seem to understand what MUDs are for.)

This is more than just a game, because many of us don't just 'play' here, but we also meet in small and large groups. We met here in the MUD and became friends in the REAL world. This is also more than just a game because the programming wizards are doing real WORK here. So when you send tells to wizards please keep in mind that your tell might disturb them. Many wizards like to chat to players and will listen to your ideas, but if a wiz tells you that he/she is working (or does not tell back) PLEASE please respect that.

Part III

Third: Sure, life is hard, MUDlife and death are hard also. Being friends means that we help each other. Some kind of help is no fun though.

Giving away quest solutions, or giving an utter novice 100.000 coins spoils the work of the wizard who programmed the quest and also the fun of the player who is helped along.

Right, some quests need a little helping with and 500 coins get the utter novice a long way. Use your brain when you help people, give them hints, not solutions.

Wizards are not allowed to help and might get problems if they do. So do what some people ask of 'Big JC' (Woopie Goldberg in 'Sister Act'): "Do not lead them into temptation." Ok?

Part IV

Fourth: Yes, there are bugs in here which might kill you or cause other problems.

- Use the 'bug' or 'typo' commands to tell the wizard in charge what the problem is.
- Use the post office to send mail to the wizard, if you know his/her name.
- Tell an Archwizard or Elder Wizard about the problem. They will know what to do about it. ('Normal' wizards might neither know how to help you, they might not be able to (because they can't change other wizards programs), or they might not be allowed to without consulting Arches/Elders.)

There are also bugs which lead to an unfair advantage. If we find out that you exploited the bug to your own advantage we will see to it that this advantage is removed (coins/exp/questpoints are taken away). If you TELL us about the bug there might be a reward though. So please, if you discover something which is not in order TELL us.

Part V

Two or more players by the same person logged in at the same time are cheating. Advantages from cheating will be removed. Apart from that this behaviour blocks login places for other players. Play fair. Exploiting a bug is not tricking the wizards, it is betraying your fellow players.

Fifth: If you feel treated unjustly by a player or wizard first talk to one of the Arches or Elder wizards. They will know what to do or inform the God Demos about it if necessary. (Demos is pretty busy managing all the aspects of the mud.)

This is it for the moment. We will add to this document if necessary. So have a look at this now and then.

Anhang D

Das Agreement

FOREWORD

This document describes the legal implications of writing code for use in TubMud and rules of conduct for handling code. This type of document is usually not necessary in leisure programming but one day there may be a problem and then it is better if things have been settled in a clear manner in advance.

It also describes the ethical rules of conduct for handling the code that you as well as others have produced. These are very different from legal rules in the copyright notice. The legal rules say that you can't sue someone just because your code pops up somewhere else in the world. The ethical rules say that you are a bastard if you take someone else's code without permission and put it up somewhere else. They also say that if you behave like a bastard you can be thrown off the game.

The reasons behind formulating things this way are several:

- Copyright issues and legal matters in this environment are extremely complicated and largely unexplored. The fact that programmers are sitting in a number of different countries, sometimes producing the code on their home machine and sometimes inside the game is one such complication. An other one is whether TubMud is to be considered as one single large program or if each object is a separate program. If the first alternative is the case it means that there is a collective copyright which belongs to all people who have written any part of the code we are using and if anyone wants to distribute any part of it he needs the consent of ALL the others. As the copyright notice is formulated it is always clear who holds the rights to a certain piece of code.
- Most of the code that is written for TubMud is the result of endeavors from a lot of people. To give the person actually writing the object all the rights would be to deny all the others theirs.
- Mud is a hobby and monetary interests, legal action and hassles about code ownership detract from the enjoyment of the hobby.

- A lot of people have an emotional investment in the code they have written and in TubMud in general. They deserve some sort of influence over the products of their imagination.
- Very drastic changes in the world detract from the players enjoyment of the game. The copyright notice allows the admins to keep the code running under all circumstances.
- It is very hard to foresee things such as net access and machine availability in the future. It should be possible to transfer TubMud to other competent hands without having to ask each and every wizard in the game if it's ok with him/her.
- It could be argued that it is possible to develop code outside TubMud and then move it here. However it is not in the interest of anyone except the coding person that he/she retains all privileges.
- Please note that the copyright belongs to the producer of the code if he/she wishes so. He/she just promises to exercise the right in certain ways. The main effects of retaining copyright is that you have RIGHT to to be known as the author of the code.

COPYRIGHT LICENSE

1. TubMud is is the name of the LPMud world run at Technische Universität Berlin, Germany. The code making up the world is termed TubMud and the society is termed Autonome Seminare in the following text.
2. All code that is produced in TubMud or imported to TubMud is subject to this license. If you have code that you for some reason can't or don't want to place under this license, don't import it.
3. All code in TubMud may be freely copied and used by anyone gaining access to it, with the exception that no code may for any reason be used for the purpose of making a monetary profit. This means that while you are allowed to run the mud, charging money from those who use it to cover the running costs, you may not make a profit from it.
4. All code in working order that is exported outside TubMud should contain this license, or a reference to it.
5. The copyright to all code that is not accompanied by an explicit copyright notice with the word Copyright, the year of creation and the name of the author is held by Autonome Seminare.
6. German law applies to this license. Among other things this means that all code is protected, that contains the author's name.

ETHICS

The following paragraphs are recommendations.

1. Put a comment containing your name, the date of creation and the fact that it was created on TubMud at the top of each file you make.
2. Put a comment with the date and information on the change every time you make a major change to one of your objects.

These are to be considered as rules of conduct. Breaking them in a major way may trigger disciplinary actions.

3. Write code that is in accordance with the rules and in balance with the game.
4. Don't copy other wizards code without permission from that wizard, the lord of that domain, or a keeper.
5. Don't export other wizards code without permission from that wizard or a keeper.
6. If you change less than half the code of an object, keep the name of the creator and add your own below.
7. If you change more than half of the code of an object, put yourself in as creator, but keep a credit to the original creator.
8. Don't place objects that differ little from the original into the game. Every creation should be in some way unique.
9. Don't import/export entire areas. TubMud should be a unique world.
10. If you make code that is not in balance with the game, is against the rules or contains bugs, the lord of your domain, an archwizard or an administrator may at any time change it. The same applies to code that has to be changed due to changes in the surroundings.
11. Changes made by others should always be documented in the code, and normally be reported to the wizard in question.
12. Code that is left unattended for a long time may be turned over to another wizard by an administrator. Unattended code in a domain becomes common domain property.
13. Administrators will only give permission to copy or distribute the code of a wizard if the wizard has been inaccessible for several weeks. The administrator will use his/her own judgement to determine if permission is warranted or not.
14. The Technische Universität of Berlin owns the machine that TubMud runs on. The society reserves the right to do anything (that does not violate the the copyright notice) it finds suitable with any information stored in the computer, but will under normal circumstances refrain from such actions without the consent of the authors.

The society is a democratic organization with bylaws, a board and regular meetings where decisions are made. The TubMud administrators, the board and the meetings are the only ones who can make decisions affecting TubMud. Individual members do not have the power to do so. Neither do individual members have the right to distribute code without permission from the wizard in question.

Anhang E

Die Schließungsmeldung von TappMud

We have noticed an increasing number of users, who obviously are not able to limit the usage of MUD to a tolerable amount of time. Users playing 6-8 hours daily are not the exception — even “records“ with more than 12 hours were noticed.

As a negative influence on the studies seems to be inevitable in these cases, we cannot take the responsibility for a usage of workstations and networks of the university any longer.

Up to now we have tolerated the MUD-Server but under these circumstances this is not possible any longer.

So this MUD-Server is out of service from now on.

To prevent our students from further addictive behaviour, we have chosen to remove all the files of this server and we are not able to provide you with copies of files from this MUD-server.

19.10.93 Dr. Juergen Kleinoeder
 <kleinoeder@informatik.uni-erlangen.de>

No drugs! No MUD-junkies! Stay clean!

Anhang F

Ein Programmbeispiel

```
/* Sharilla's special gift */
/* 30-09-93 */
/* test test.... *shiver* */

#define AUTH_FILE "/players/sharilla/data/scarfowners"

status worn;
status authorized(string str);

status id(string str)
/* wie lassen wir uns denn anreden?? */
{
return str == "scarf" || str == "silk" || str == "label" || str == "corner";
}

string short()
{
if (worn==0)
return "a black silken scarf";
else if (worn==1)
return "a black silken scarf (worn around the neck)";
else
return "a black silken scarf (worn around the arm)";
}

string long(string str)
{
if (str == "label" || str == "corner")
return "It seems to be a compass, pointing in eight directions at the same\n"
+"time. Confusing imaginations of pure chaos come to your mind.\n";
else
return "The black square piece of silk in your hand reminds you of fragile\n"
+"spiderwebs. No doubt, this is real elvenware! You recognize the\n"
+"fragrance of a heavy perfume. A small label in one corner\n"
+"attracts your attention.\n";
}
```

```

void reset(int arg) /*bei jedem reset... */
{
    if(arg==0)
        worn=0; /*am Anfang wird das Tuch nicht getragen*/
    return;
}

status drop(status silently)
/* beim quitten wird silent entsorgt, gedropet wird nich! */
{
    if (query_verb() == "give")
    {
        worn=0;
        return 0;
    }
    if (!silently && query_verb() != "quit")
        write("Your heart seems to break, when you think of dropping Sharilla's gift!!\n");
    return 1;
}

mixed *query_auto_load()
/* wir sind ja sowas von autoloading.. */
{
    return ({ program_name(this_object()), worn});
}

void init_arg(mixed * args)
/* je nach Inhalt autoloading sind wir getragen oder nich */
{
    if (args!=0)
    {
        worn=args[0];
        write("Your black scarf nearly seems to purr as you wear it again. \n");
    }
}

status get() /*getten erlaubt*/
{
    return 1;
}

void init()
{
    if (this_player()==environment(this_object()))
    {
        if (!authorized(this_player()->query_vis_name()))
        {
            write("You are not authorized to own Sharilla's wonderful scarf.\n");
            call_out("self_destruct",0,this_object());
            return;
        }
    }
}

```



```

add_action("smell_scarf", "smell");
add_action("tie_scarf", "wear");
add_action("tie_scarf", "tie");
add_action("remove_scarf", "untie");
add_action("remove_scarf", "remove");
add_action ("help_scarf", "help");
}
}

void self_destruct(object scarf)
{
    destruct(scarf);
}

status smell_scarf(string str)
{
    object sharilla;
    sharilla=find_player("sharilla");
    if (!str || str != "scarf")          /*nur smell oder smell irgendwas */
    {
        notify_fail("Where do you want to smell?? \n");
        return 0; /* return nase?? */
    }
    else /*ordentlich geschnueffelt*/
    {
        write ("You hold the silk close to your face and breathe it's fragrance.\n"
            +"Everything around you seems to fade away. You remember that only\n"
            +"one person in this world uses this magical perfume and are\n"
            +"overwhelmed by the desire to search the presence of the dark\n"
            +"princess you see clearly before you in your imagination.\n");
        say(this_player()->query_name()+" seems to get a lot of satisfaction by
            smelling at a small black scarf.\n");
        if(sharilla) tell_object(sharilla, this_player()->query_name()+" dreams of
            you.\n"); /* hihi */
        return 1;
    }
}

status tie_scarf(string str)
{
    string*woerter;
    if (str == 0)
    {
        notify_fail("What do you want to wear?? \n");
        return 0;
    }
    woerter=explode(str+" ", " "); /*blank an str, sonst fehlt das ende*/
    if (sizeof(woerter)<3 || woerter[0] != "scarf" || (woerter[1]!="to" &&
        woerter[1]!="around") || (woerter[2]!="neck" && woerter[2]!="arm"))
    {
        /*tie-to, wear-around sind ok, alles andere rott */
        notify_fail("What do you want to wear around what?? \n");
        return 0;
    }
    else if (worn==0) /*dann wirds getragen */

```

```

{
say(this_player()->query_name()+" ties a wonderful black scarf "
    +woerter[1]+" "+this_player()->query_possessive()+" "+woerter[2]+".\n");
if (woerter[2]=="neck")
{
write ("You tie the wonderful cool silken scarf around your neck. \n");
worn=1;
}
else
{
write ("You tie the scarf around your arm. \n");
worn=2;
}
return 1;
}
else
{
write("You already wear it.\n");
return 1;
}
}

status remove_scarf(string str)
{
if (!str || str != "scarf")
{
notify_fail("What do you want to remove?? \n");
return 0;
}
else if (worn != 0)
{
say(this_player()->query_name() + " removes "
    + this_player()->query_possessive() + " black scarf. \n");
write ("You remove the wonderful black scarf, a feeling of loss comes to you. \n");
worn=0;
return 1;
}
else
{
write("You aren't wearing it now. \n");
return 1;
}
}

status help_scarf(string str)
{
if (!str || str != "scarf") /*nur help oder help irgendwas */
{
notify_fail("Try to make explicit, what kind of help you need. \n");
return 0; /* return nase?? */
}
}

```

```

else
{
write ("Trie to <tie> the scarf around your arm or neck and <wear> it or \n"
      + "<smell> the interesting perfume\n"
      + "You may also <remove> or <untie> it.\n");
return 1;
}
}

status authorized(string str) /*darf derjenige mein scarf ueberhaupt haben?*/
{
string names;
names=read_file(AUTH_FILE,0,-1); /* ab byte 0 und maxanzahl bytes*/
if (member_array(str,explode(names,"\n"))!= -1)
return 1; /*is authorisiert*/
else
return 0; /*buuuh! fiese moeb! */
}

```