

Nemesis - ein typisches MUD im Deutschland der Neunziger

Astrid Jekat, 2014

Was ist ein MUD?

Definition:

Multi-User Dungeon, auch **Multi-User Dimension** und **Multi-User Domain**.

MUDs sind textbasierende frühe Onlinespiele.

Es sind virtuelle Welten mit zahlreichen Spielern, die interagieren können. Diese Welten verbinden Elemente von Rollenspiel, Abenteuerspielen, Duellen, Rätsel, interactive fiction, und online Chat. Spieler lesen die Beschreibungen von Räumen, Gegenständen, anderem Spieler, NPCs, sowie von Handlungen/Ereignissen in dieser virtuellen Welt. Die Interaktion mit dieser Welt sowie anderen Spielern erfolgt über Kommandos, die einer natürlichen Sprache entnommen sind. Die Blütezeit der textbasierenden Onlinespiele war in den 1980er/90er Jahren, ihre grafischen Abkömmlinge erfreuen sich heute noch grosser Beliebtheit.

Viele MUDs folgen dem Muster von Fantasyrollenspielen wie D&D, in denen man sich mit erfundenen Rassen und Monster auseinandersetzt, um Fertigkeiten, Gegenstände, oder Titel zu erwerben. Um dies zu erreichen und die eigene Rolle (den „character“) weiterzuentwickeln, erforscht man unbekannte Gegenden, kämpft mit Ungeheuern, erlebt Abenteuer, und erschafft dadurch eigene Geschichten.

Der Aufstieg zu der besonderen Rolle des „Wizards“ im MUD ist möglich – dies sind hochrangige Spieler, die erweiterte Befugnisse erhalten. Wizards sind Coder im Spiel, und können administrative Aufgaben ausüben, ähnlich einem heutigen Gamemaster.

Es gibt verschiedene Spielarten: Hack&Slash (Abenteuer bestehen/Monster besiegen), PvP (Spieler im Kampf mit anderen Spielern), reines Rollenspiel und Darstellung, educational (z.B. MUDs in Fremdsprachen, vgl. das Latein MUD), sowie Chat MUDs.

Aus diesen frühen Textspielen entstanden später sog. „grafische MUDs“ wie Habitat, Everquest, Ultima Online. Der Begriff wandelte sich dann zu MMORPG (*Massively Multiplayer Online Role-Playing Game*). Letztendlich führt die Entwicklung direkt zu unseren modernen Zeitvernichtern wie WoW, Starwars Old Republic etc., sowie zu neuen Spielplattformen wie etwa dedizierten Spielkonsolen (zuerst SEGA Dreamcast, dann v.a. Sony PlayStation), und als neueste Entwicklung zu Apps auf fast jedem Smartphone.

Wie entstanden MUDs? Welche Einflüsse gab es?

Vorläufer/Einflüsse:

- Tolkienromane.
- Pen&Paper Tischrollenspiele (D&D).
- Interactive Fiction bereits in Buchform, aka Spielbuch/game book, bei denen Leser als *Spieler* direkten Einfluss auf die Handlung des Buchs nehmen kann. Erste game books wurden bereits in den Siebzigern veröffentlicht. (*Sugarcane Island / Insel der 1000 Gefahren*, 1976. Ab1981 die an D&D Spielprinzipien und Kampfsystem angelehnte *Fighting Fantasy* Reihe.) Andererseits wurden game books auch als Spinoffs von single user adventures (v.a. von Zork) erstellt.

Single User Text-Based Role-Playing Adventures:

Colossal Cave Adventure/Adventure auf PDP-10 1976:

Der Programmierer **Will Crowther** arbeitete in den 70ern mit an der Entwicklung des ARPANETs, und war gleichzeitig ein passionierter Höhlenforscher. Er entwickelte ein Abenteuerspiel, das auf der Mammoth Cave im US Bundesstaat Kentucky basierte (angeblich um es mit seinen Töchtern zu spielen). Schon in dieser Version findet man eine magische Brücke und einen axtwerfenden Zwerg.

Basis:

FORTRAN / FORTRAN IV/ FORTRAN 77

Problem: Gesamtes Spiel mit allen Daten ins RAM geladen, rein prozedurale Sprache, schwer von PDP-10 weg portierbar – später C unter Unix, sowie kleinere Rechner wie PET, Commodore 64, ZX-81, Apple II und TRS-80).

Heute Teil der BSD-Unix-Distributionen sowie Linux.

Don Woods an der Uni Stanford entwickelte das Spiel weiter und fügte typische Elemente aus Tolkienromanen hinzu wie Elfen und Trolle.

Der Begriff des Wizards als Ziel für den Spieler taucht erstmals auf bei der Vermarktung des Spiels unter dem Namen „The Original Adventure“ für Heathkit und IBM-PCs. Nach kompletter Lösung des Spiels wurde ein Code ausgegeben, mit dem man ein „Certificate of Wizardness“ mit den Unterschriften von Crowther und Woods bestellen konnte. Zum ersten Mal konnte man „aufsteigen“.

Basis:

Erste Ansätze spielspezifischer Sprachen: „A-Code“, eine Art Präprozessorsprache, die durch ein eigens in F77 geschriebene Anwendung übersetzt wurde in eine Textdatenbank und ein Pseudobinary, das durch eine Art F77 „executive“/quasi kernel ausgeführt wurde. *Hier sieht man auch erste Ansätze von Aufteilung der Spielinfrastruktur.*

Zork 1977-79 (Zeitweilig Dungeon)

„Zork“ war ein Phantasiewort von MIT Entwicklern dieser Zeit, um unfertige Programme zu bezeichnen, die noch nicht auf dem System installiert waren. Die erste Version von Zork wurde 1977–1979 von den MIT Entwicklern **Tim Anderson**, **Marc Blank**, **Bruce Daniels**, **and Dave Lebling** auf einer DEC PDP-10 in MDL geschrieben.

Basis:

MDL = LISP Abkömmling, prozedural, aber wichtige Datentypen wie Arrays und benutzerdefinierte Datentypen. Lokaler und globaler Scope von Variablen definierbar.

Weiter entwickelter Parser für Kommandos: "hit troll" → „hit troll with elvish sword“.

Spielkonzept : Navigation, Gegenstände sammeln, Rätsel lösen, Storyline folgen, hack&slash, Fantasyelemente, Humor. Sterblichkeit des Spielers als Kontrapunkt zur Lösung des Spiels. Berühmter Catchphrase: "You are in a maze of twisty little passages, all alike"

Letztendlich entstand aus dieser Gruppe die berühmte Firma Infocom, welche Zork auf diverse Mikrocomputer (Apple II, [Commodore 64](#) / [Commodore Plus/4](#), [Atari 8-bit](#) Familie, [TRS-80](#), [CP/M](#) Systems, [IBM PC](#)) portierte und vertrieb.

Trotz der spieleigenen Sprache und virtual machine musste immer das gesamte Spiel ins RAM geladen werden, um zu laufen. Closed Source, natürlich.

Grosse Programme auf kleine Maschinen:

Joel Berez und Marc Blank entwickelten dazu die Z-machine, eine virtual machine auf der Zork mit eigenen Z-code story files lief. Dies erleichterte die Portierung auf die zahlreichen inkompatiblen Plattformen sehr, da nur eine Z-machine für jede Plattform implementiert werden musste (statt native code für jedes Spiel auf jeder Plattform). MDL wurde reduziert zu ZIL ("Zork Implementation Language") -> kompliziert zu Z-code. Der ZIP (Zork Interpretive Program) Interpreter erzeugte den Maschinencode der Plattform.

(Rogue, Hack/Nethack: hingegen entwickelten bereits auf Textkonsolen Maps und Raumdarstellungen statt Textbeschreibungen. Gesamte Gamemap in jeder Instanz komplett neu random erstellt versus meiste feste Raumstruktur in zork und Abkömmlinge.)

Erste online MUDs:

Erweitertes Spielkonzept:

Mehrere Spieler können erstmals gleichzeitig in derselben Spielinstanz einloggen und miteinander interagieren. Chat, Teams und PvP- Kämpfe werden möglich.

MUD (Multi-User Dungeon)

Roy Trubshaw, ein Student an der englischen Uni Essex, entwickelte in **Anlehnung an die *Dungeon* genannte Variante von Zork 1978** das erste Echtzeit Multiplayer Spiel namens.... *MUD* bzw *MUD1*, *ESSEX MUD* (später auch der Klon *Essex MIST*). Dies prägte den Begriff, der bis heute verwendet wird!

Es wurde in MACRO-10 auf PDP-10 entwickelt, später portiert auf BCPL (C Vorläufer). Das Spiel lief bis 1987 im Universitätsetz Essex, später mit Zugang über das britische wissenschaftliche (X.25 basierende) JANET. Richtig beliebt wurde es durch 1980 durch den Zugang zum ARPAnet.

!! In diesem PvP Spiel kann man sich erstmals zum besonderen Status des unsterblichen Wizards hocharbeiten, inkl. Macht über andere (sterbliche) Spieler. Problem war aber das völlige Fehlen von Wizardregeln / übergeordneter Instanz, was zu chaotischen Zuständen bis hin zum willkürlichen Umgang mit Spielern führte!

1984 konnten Trubshaw und Bartle es unter dem Namen MUD1 bei CompuServe USA und CompuNet Großbritannien platzieren, wo es enormen Zulauf erhielt und später als MUD2 auch über die britische Telecom erreichbar wurde. MUD1/2 gilt als das Ur-MUD schlechthin. Eine emulierte Version kann übrigens noch heute gespielt werden.

Weitere MUDs in anderen Netzen/Systemen:

CompuNet, British Telecom's Prestel und Micronet Netzwerke, später AOL stellten weitere MUDs zur Verfügung wie *SHADES*, *GODS*, *MUCK* (Multi User Computer Kingdom) und das berühmte *Federation I* und *Federation II* (bei AOL: "*Federation: Adult Space Fantasy*").

(Technisch: Meist auf Mainframes)

BBS / Mailboxnetze:

1978 schrieb Alan Klietz in den USA (Minnesota) das Spiel *Milieu* auf einen CDC Cyber 6600 Mainframe. 1983 portiert auf einen IBM XT Rechner und zu *Scepter of Goth* umbenannt, konnten sich bis zu 16 Spieler gleichzeitig per Modem einwählen. Dies war eines der ersten kommerziellen MUDs (sprich ausserhalb des akademischen Systems) überhaupt.

1985 betrieb Pip Cordrey in England ein Mailboxsystem, auf dem ein MUD1 Klon namens

MirrorWorld lief.

Die Portierung verschiedener MUDs (z.B. *Swords of Chaos*, *Trade Wars*) auf MS-DOS Systeme sowie andere Mikrocomputer führte zu grosser Verbreitung in Mailbox/BBS-Netzen bis weit in die 90er.

AberMUD

Der erste Mud mit veröffentlichter codebase, entstanden an der University of Wales in der Stadt Aberystwyth. Bisher musste man sich Zugriff auf oder Kopien des source codes eines Spiels erschleichen.

Alan Cox kannte das originale ESSEX MUD, und schrieb seine Fassung eines Spiels zunächst (1987) in B auf Honeywell L66 mainframe under GCOS3/TSS, ein Jahr später C auf diversen Unix Plattformen. Spätere Versionen liefen anderen britischen Unis (Southampton, Leeds, London) und wurden in die USA exportiert.

AberMUD löste die Entwicklung von berühmten codebases aus: TinyMUD, DikuMUD, und LPMud.

Es laufen noch etliche AberMUDs (*Northern Lights*, *Infinity*) online!

TinyMUD:

1989 modifizierte **James Aspnes** das Spiel *Monster* und baute ein MUD mit cstatt Punktesammeln (eine spätere Version führte erstmals ein whisper Kommando ein). Als Neuerung gibt es Spielern die Möglichkeit, selber Räume, Gegenstände, Rätsel für andere Spieler zu erstellen. Änderungen am Spiel werden sofort aktiv.

LPMUDs:

Der Schwede Lars Pensjö hatte sowohl AberMUD als auch TinyMUD ausführlich gespielt. Er war bereits berufstätig als Entwickler von Unixtreibern und Festplattenfirmware, als er die Sprache LPC für LPMUDs entwickelte. Das Ziel dieser dedizierten Sprache war es, möglichst speichereffizient die Flexibilität und Erweiterbarkeit von TinyMUD mit den Spielprinzipien und Abenteuern von AberMUD zu verbinden. Er erstellte in LPC den sog. LPMud, die Codebasis für ein Spiel mit der benötigten internen Struktur, Objekten, und einem rudimentären Spiel.

Außerdem wollte er nicht der alleinige Entwickler des Spiels sein: *'...I didn't think I would be able to design a good adventure. By allowing wizards coding rights, I thought others could help me with this.'* Aus dem einsamen Spieleentwickler wurde eine grosse Gemeinde von Codern mit verteilten Aufgaben/Bereichen im Spiel, die nicht alle tiefen Einblick in Programmierlogik haben müssen. Stattdessen können sie sich auf die Erweiterung des Spiels selber konzentrieren.

In den frühen 90ern wurde LPMud rasch zur beliebtesten Grundlage von Onlinespielen. Die Entwicklung wurde von Jörn "Amylaar" Rennecke, Felix "Dworkin" Croes, Tim "Beek" Hollebeek und Lars Düning weitergeführt. Abkömmlinge von LPC sind u.a. LDMud (Lars Düning, 1997), MudOS, DGD, SWLPC, FluffOS, und die Programmiersprache Pike.

Lars Pensjös originales LPMud namens *Genesis* läuft immer noch online!

Eigenschaften:

Leichtere Veränderbarkeit des Spiels, Hinzufügen beliebig komplexer neuer Teile zur Laufzeit! Aufgrund der Objektorientierung muss nicht das gesamte Spiel ständig geladen sein (wie z.B. bei DikuMUDs).

Struktur

Aufteilung in gamedriver – mudlib!

Ein in C geschriebener **gamedriver (LPMud driver)** übernimmt Aufgaben, die man heutzutage als die einer Virtual Machine bezeichnen würde. Der gamedriver stellt das Betriebssystem für das Spiel zur Verfügung, und ist gleichzeitig der Interpreter des LPC Codes.

Außerdem schützt dieser neue quasi Layer den Server vor Programmierfehler im Spiel – bei loops oder infinite recursions werfen Objekte Fehler oder laden nicht; schlimmstenfalls stürzt der gamedriver ab.

Die **mudlib** stellt die grundlegenden Elemente für das Spiel zur Verfügung. Die wichtigsten MUD Objekte sind u.a. rooms (Umgebungen), Waffen, Rüstungen, und interaktive Objekte (Player, NPCs/Monster). Die mudlib liefert die Grundform (Prototypes) dieser Objekte mit Basiseigenschaften und -fähigkeiten, die an andere Objekte vererbt und dort durch Wizards/Coder modifiziert werden können. Z.B. `/lib/room`, von dem alle Rooms im Spiel abgeleitet werden.

In verschiedenen MUDs wird häufig die mudlib modifiziert und erweitert.

MUD1 had a mudlib, but it was an adaptation of the BCPL input/output library and therefore was at a lower level than today's mudlibs. The modern usage of the term was coined independently by LPMUD.

LPC, eine MUDspezifische Programmiersprache

Frühe Programmiersprachen von Spielen, Grenzen:

Prozedurale Sprachen, langer Code, Spiel musste komplett ins RAM geladen werden, kaum veränderbar zur Laufzeit.

Organisatorisch: Einige wenige zentrale Entwickler, code war closed source.

LPC – dynamische Spielentwicklung:

LPC ist am besten dafür bekannt, dass Objektorientierung in die MUDprogrammierung eingeführt wurde. Sie ist nicht die erste OOP Sprache, aber die erste ausdrücklich für die Spieleentwicklung.

Objektorientierung

LPC ist von plain C abgeleitet (Syntax!) und durch Objective C / C++ beeinflusst.

Beinahe alles im Spiel (Räume, Gegenstände, Waffen, sogar die Player und Wizards) ist ein Objekt.

Hält Grundsätze von OOP ein, aber vereinfacht im Vergleich zu modernen OOP Sprachen.

1. *Abstraktion: Das sichtbare Verhalten des Objects legt nicht offen wie genau diese Fähigkeiten implementiert worden sind*
2. *Encapsulation/Datenkapselung: das Verbergen von Implementierungsdetails. Auf die interne Datenstruktur kann nicht direkt zugegriffen werden, sondern nur über definierte Schnittstellen (auf*

- welche Weise mit dem Objekt interagiert werden kann: In LPC = functions). Objekte können den internen Zustand anderer Objekte nicht in unerwarteter Weise lesen oder ändern.
3. Modularität: Viele kleinere Objekte..
 4. Hierarchie/Vererbung: von Eigenschaften zwischen Klassen/Objekten.
 5. Polymorphie: Verschiedene Objekte können auf die gleiche Nachricht unterschiedlich reagieren.

ABER: In LPC wird auf die Deklaration von Klassen gänzlich verzichtet. Stattdessen werden neue Objekte von bestehenden Objekten, den sogenannten Prototypen/Master Object in der mudlib, abgeleitet (Befehl: clone). Die Attribute und Methoden des Prototyps kommen immer dann zum Einsatz, wenn sie im abgeleiteten Objekt nicht explizit überschrieben wurden. Dies ist vor allem für die Entwicklung kleinerer Programme von Vorteil, da es einfacher und zeitsparend ist. Anders strukturierte MUDs wie z.B. DIKUMuds laufen effizienter, können aber ohne tiefgehende Programmierkenntnisse nicht angepasst werden.

Nemesis: Geschichte

Import aus UK:

1990 kehrte „Barny“ (M. Arbesmeier), ein Diplomand der TUM, von einem Auslandsaufenthalt in den UK zurück mit einer Kopie der ersten Version von LPMUD und LPC im Gepäck.

Start am Zenger Lehrstuhl:

Er zeigte das Spiel seinem Kommilitonen „Snake“ (R. Gebhart), der nebenbei Sysadmin an Lehrstuhl Zenger war. Er installierte Januar 1991 den GD inkl. mitgebrachter MudLib und ein paar Castles auf einer DECstation 2100 mit 28Mbyte RAM unter Ultrix 4.1 und startete den LPMUD gamedriver. Damit nahm das Unglück seinen Lauf. :-)

Nemesis, das Fantasy MUD war von da an auf der Maschine dszenger9.informatik.tu-muenchen.de (131.159.8.67, port 2000) am Zenger Institut der TU München erreichbar.

Herkunft Name Nemesis:

Es sollte eine Parallelwelt im Spiel erzeugt werden, aber der Begriff der Gegenerde war schon durch Autoren mit Copyright belegt (Terry Pratchett/Scheibenwelt etc.) belegt. Spontan wählten die frühen Wizards (Junky, Nik, Storm, Snake) den Begriff „Nemesis“ für seinen leicht unheimlichen Klang. Eine spätere Recherche ergab, dass dies ein pseudowissenschaftlicher Begriff der 80er für einen Begleiter der Sonne (Paralleluniversum, periodische grosse Katastrophen) :-).

Blütezeit von Nemesis:

Februar 91 bekommt die Maschine, auf der Nemesis läuft, internationalen Internetzugriff (1992: 2Mbit/s X.25-basierter DFN/WiN-Anschluss des LRZ). Spieler aus der ganzen Welt loggen ein, spielen sich hoch, und erlernen dann als Wizards OOP.

1993 werden ein gopher server und ein SMTP gateway (interne Nemesis mail → externer SMTP server) entwickelt. Später Implementation eines HTTP servers in LPC.

Thema/Umwelt:

Technologie, Waffen, Erscheinung ist auf Fantasy ala Tolkien beschränkt, aber Ausnahmen sind erlaubt (z.B. Zeitsprung / magisches Portal in ein eng begrenztes Gebiet etc.).

Die Welt ist eine Kugel mit etlichen Inseln im Weltmeer, die durch Schiffslinien erreichbar sind. Währungen sind im Ansatz da (noch nicht aktiv), aber kein ökonomisches / ökologisches System.

Ziel des Spiels ist es, durch Sammeln von Erfahrung (hack&slash, Rätsel) und Lösen von Quests in den obersten Spielerrang aufzusteigen. Man kann dann auch die Aufnahmeprüfung zum unsterblichen Wizard machen, um das Spiel weiter zu entwickeln.

Wachstum

Apr 1993: Spielerzahl: 4940 (Total), 1249 (Active), 3691 (Inactive)

Active Wizards: 196, Players: 1053

Login numbers:

during restriction time: max 15, av 5 players

during free access time: max 35, average 20 players

Top Countries:

| | |
|-------------------|---------------|
| 551 USA | 5 Spain |
| 308 Germany | 5 Sweden |
| 51 United Kingdom | 4 Austria |
| 32 Canada | 3 Finland |
| 28 France | 2 Belgium |
| 26 Australia | 1 Iceland |
| 15 Italy | 1 Japan |
| 5 Netherlands | 43 Unresolved |

Top Universities:

| | |
|-----------------|---------------|
| 74 Ohio-State U | 32 Warwick |
| 54 Munich TU | 23 Mainz U |
| 47 Buffal U | 21 Dayton U |
| 30 Albany U | 21 Weber U |
| 24 Strasbourg U | 20 Erlangen U |

Features:

Technik/Version/Modifikationen:

Original

Driver: LPmud 3.1.2, stark modifiziert : Sicherheitsmassnahmen, Lauschen auf mehreren Ports, outgoing Verbindungen auf beliebigen TCP-Ports (ermöglicht LPC-Gopher-Server, LPC-Webserver, in/outgoing SMTP-Mail).

Mudlib: former 2.4.5, stark modifiziert (neue Kommandos und Standardobjekte)

Hardware: DEC 2100, Ultrix 4.1

Address: dszenger9.informatik.tu-muenchen.de (131.159.8.67) 2000

„Open to players 5pm - 9am CET (central european time).

Wizards can log in any time.“

Used disk space: 60 Mbytes

Reboots: automatically once a week (Sunday night), due to special occasions (driver updates etc) twice a week on average

Damalige Beschränkung der Anzahl von Playern üblich, um Systemressourcen/Bandbreite zu sparen → Max Players: 20

Heute

Hardware/OS: virtualisiertes FreeBSD

Adresse: Telnet - mud.nemesis.de 2000 HTTP – <http://nemesis.de>

Max Players: 50

Entwicklung des Player characters:

Charaktererstellung einfach durch Login, Name/PW wählen. Jeder Spieler kann jeden anderen sehen. Kommunikation: say, tell, shout (cost: 50 sp), boards, mail. Shoutkosten verringern den Lärm im Spiel erheblich. (Wizards können durch unsichtbar werden und „earmuffs“ komplette Stille für Debugging erreichen.)

Spielen ist die Hauptbeschäftigung des Players, aber das Chatten wird gefördert (Handel zwischen Playern findet statt, kooperative Quests sind in Planung). Dies inspirierte sogar später die Entwicklung des LPC-basierenden Chatsystems *Psych* durch den Wizard „Lynx“.

Der Spieler steigt zu höheren Levels auf durch Sammeln von experience points (Gegner besiegen, Rätsel lösen, besondere Gegenstände finden, sowie das Lösen der verpflichtenden Quests, meist eine Kombination aus allem). Bei Erreichen von level 20 kann der Spieler eine Prüfung (Verhaltenskodex) ablegen und durch Admins/God zum Wizard, also einem neuen Entwickler gemacht werden.

Kampfsystem: Komplette neu erstellt (durch Lynx), läuft nach Beginn automatisch ab (aber Flucht fast immer möglich), keine Definition von Trefferbereichen, aber multiple Angreifer sowie Teilen von XP möglich.

Player“soul“, die allerlei Gefühle/Handlungen für Rollenspiel ausdrücken kann (cry, laugh, sing, bow, dance), kombiniert mit vielen Adverbien (happily, sadly, wildly, demonically etc.)

14 Quests (zwingend nötig für Aufstieg) sowie allerlei freie Tasks/Miniquests für XP.

Wettersystem: Verschiedene Wetterlagen inkl. Effekte auf Player (z.B. werden Kerzen/Fackeln im Regen nass), Tag/Nachtzyklus (mit echter Dunkelheit), eigene Zeit (1 Nemesisstag = 4 Echtzeitstunden bzw 8fache Beschleunigung), eigener Kalender (derzeitiges Nemesisjahr ist 1411).

System von 13 Schiffen zwischen den Inseln, mit Reisezeiten auf dem Schiff, festen Routen und Abfahrtszeiten.

Administration:

Wizard Hierarchie: je höher, desto mehr file access und Verantwortung:

1 God (höchster Coder, verantwortlich für den gamedriver)

5 Administrators (hohe Coder, fällen wichtige Entscheidungen für das Spiel)

3 Archwizards

14 Domain Lords für 11 Domains (seit Feb 1992). Domain Lord betreut eine Gruppe von Wizards, die gemeinsam ein Gebiet mit festgelegtem Thema coden.

Viele Wizards: Coder für kleinere eigene Bereiche, meistens einer Domain zugeordnet.

Wizards sind verpflichtet, Spielern in Schwierigkeiten zu helfen, und dürfen allgemeine Auskünfte geben – aber keine Lösungen verraten!

Wizard Tools:

Dokumentation: man pages, example rooms, tutorials.
Generische Objekte/Vorlagen. Entwicklungs-/Debug tools.

Zuständigkeit Code:

Gamedriver: 3 Wizards aus München

Mudlib: 5 Wizards aus München, 1 aus USA

Welt: Alle vollwertigen wizards (zur Blütezeit bis zu 200 Aktive, jetzt eher 20).

Eigene Gebiete („castles“) werden durch die Domain Lords freigegeben.

Snooping is erlaubt, bei Misbrauch jedoch Sanktionen.

Datenübertragung via mtp, ftp, neuerdings MudFM via Webserver.

Ethik:

PvP ist nicht per default möglich, ebenso ist der aktive Eingriff Wizards ins Spiel (inkl. Angriff/Heiling/Beschenkug/Töten von Spielern) streng verboten. Beides soll die Exzesse, wie sie im ESSEX MUD üblich waren, verhindern.

Todesfallen sind im Spiel verboten, es muss immer einen Weg raus geben!

Jedoch können Spieler sich gezielt als Playerkiller/Desperados eintragen lassen – und es wurde auf sie ein Kopfgeld/Bounty ausgesetzt! Ausserdem greift sie dann jeder guard am Hafen an.

Zukünftige Wizards müssen sich an die Verhaltensregeln halten (kein Eingreifen, Töten, Helfen/Behindern, freundlich sein). Sie können überwacht und bestraft werden, bis hin zum Verlust des Wizards und Sperrung.

ASCII Art

Wie bei viele MUDs dieser Zeit wurden in liebevoller Kleinarbeit ASCII Bilder erstellt, die dynamische maps/Karten, Brettspiele (Go etc.), besondere Umgebungen, darstellten. Nemesis hat ca. 10 verschiedene ASCII-Bilder als Login screen zur Verfügung, die zufällig ausgegeben werden.

Warum Englisch?:

Das originale LPMUD inkl. aller Kommandos war in Englisch verfasst worden → Aufwand der Umstellung. Deutschsprachige MUDs entstanden erst allmählich.

Des Weiteren war der Parser des gamedrivers nur 7bit clean, Umlaute waren nicht vorgesehen. Eine wichtige Eigenschaft der englischen Sprache – sie ist flexionsarm! Der Parser musste nicht mit deutscher Grammatik, veränderten Schreibweisen bei Beugung etc. umgehen.

In den späten 80ern/frühen 90ern war Englisch selbst in Deutschland noch die Sprache des jungen Internets, u.a. da weite Teile Zugangs immer noch aus akademischen Netzen stattfand. Noch dazu war Nemesis wegen der relativ guten Anbindung auch aus dem Ausland in sinnvoller Geschwindigkeit erreichbar und rasch auch bei britischen und amerikanischen (> Deutschland!) Spielern beliebt.

Die Chatsprache unter Spielern/Wizards darf jedoch frei gewählt werden.

GESCHICHTE NEMESIS FORTGES:

Konflikte und vorübergehendes Ende

Es gab immer wieder ein Tausziehen, ob das Spiel zuviel Ressourcen verbraucht versus dem Lerneffekt der Wizards.

Januar 1994 veranlaßt die Verwaltung des Leibnitz Rechnzentrums mit der Begründung, es werde zuviel Bandbreite verbraucht, die Schliessung von Nemesis. Dies löst ein Medienecho, das jedoch nichts bewirkt. War dies das Ende? Nein! Denn es gab noch ein sorgfältig gehütetes Backup auf Band.

Neustart 2001, Plattenverlust, Neustart 2015

Mehrere Versuche, das Spiel wieder an der TU hochzuziehen, schlagen fehl.

März 2001 wird das Spiel auf privater Hardware des MUD Gods Snake zuhause von Band hochgezogen – diesmal auf einer DECstation5000 unter NetBSD. Die neue Ära von Nemesis beginnt mit Treffen, Aktualisierung des gamedrivers Homepages für Wizards und Domains.

2009 gibt die Platte des Servers bei einem Stromausfall den Geist auf.

Februar 2015 wird der gamedriver auf FreeBSD portiert und läuft seitdem in einer virtualisierten Umgebung.

Vorstellung auf dem VCFE auf zeitgemässer Hardware

Für diese Veranstaltung wird Nemesis nochmal auf originaler Hardware (DECstation5000) unter NetBSD ausgestellt.

Login / MUD Clients:

Telnet ist die ursprüngliche und immer noch einfachste Methode, ein MUD zu erreichen.

(*putty* für Windows tut auch.)

Clients (command history/expansion, shortcuts, scripting, messaging) für Unix/Linux/Windows, teils Mac:

powwow

tintin++ (Automapping! WinTin++. Mac Version)

Tinyfugue (tf)

Aufwendiger (Farben, Mapping, GUI etc.), auch alle OSes:

Mudlet

zMUD (nur Windows), erstellt Karten!

Beispiel: LPC Code – Sichtbares im Spiel

```
//////////////////////////////////// CODE //////////////////////////////////////
#include <room2.h>
reset(arg) {
    if (arg) return;
    set_short("nemesis village");
    set_long(
        "You are at an open green place south of the village church.\n"+
        "You can see a road further to the east and west.\n"+
        "To the south is the local cemetery.\n");
    set_exits( ({
        "room/church", "north",
        "players/kiri/c/gate", "south",
        "d/city/outer/stone0", "west",
        "d/city/outer/nemesis4", "east"
    }));
    set_items( ({
        "road", "A small and dirty road through the village",
        "church", "The church is a majestic bulding",
        "cemetery", "It looks gloomy in there"
    }));
    set_weather(LIGHT, OUTSIDE, STANDARD);
    move_object(clone_object("obj/lantern"),this_object());
    move_object(clone_object("players/dixie/examples/sundial"),this_object());
}
//////////////////////////////////// IM SPIEL //////////////////////////////////////
```

> look

You are at an open green place south of the village church.

You can see a road further to the east and west.

To the south is the local cemetery.

There are four obvious exits: north, south, west and east.

Astrid the small adventuress (neutral).

A leaflet.

A sundial.

A lantern.

//////////////////////////////////// IM SPIEL //////////////////////////////////////

> look at lantern

It seems to turn on and off automagically.

Currently it is off.

//////////////////////////////////// IM SPIEL //////////////////////////////////////

> look at sundial

It is half past 12 o'clock in the afternoon.

//////////////////////////////////// IM SPIEL //////////////////////////////////////

> look at cemetery

It looks gloomy in there.

//////////////////////////////////// IM SPIEL //////////////////////////////////////

> look at church

The church is a majestic bulding.

//////////////////////////////////// Objekt 'leaflet' steht nicht im room code – IM SPIEL //////////////////////////////////////

> take leaflet

You take a leaflet.

//////////////////////////////////// IM SPIEL //////////////////////////////////////

> look at leaflet

Visit the new Nemesis Information Center.

Right east from the church.

//////////////////////////////////// Objekt 'Astrid' steht nicht im room code – IM SPIEL //////////////////////////////////////

> look at astrid

Astrid is in good shape.

Astrid is a Citizen of Skara Brae.

Astrid is carrying:

A wet stick.

A golden apple.

True seeing.

A golden watch.

A little blue dragon.