

Szenarion
Das
Rollenspiel
Software - Paket

AMIGA
Version 1.03
von
S. Waschk

19. August 1995

Inhaltsverzeichnis

1	Distribution	3
2	Allgemeine Informationen	4
2.1	Warum brauche ich das Rollenspiel-Software-Paket?	4
2.2	Vollversion	5
2.3	Installation	5
2.4	Input	6
3	Warnung	7
4	Arbeitsablauf	8
4.1	Meister-Tools	8
4.2	Allgemein	8
4.3	Szenario erstellen	9
4.4	Kurzübersicht	11
5	Der Rollenspiel-Feld-Editor	12
5.1	Grundlagen	12
5.1.1	Maßstab	13
5.1.2	Die Modi	13
5.1.3	Objekte	13
5.2	Funktionstasten	14
5.3	Grafische Fenster-Oberfläche	18
5.4	Tips zur Felderzeugung	20
6	Rollenspiel-Kampf-Simulator	21
6.1	Arbeitsweise	21
6.1.1	Die Charakterfenster	22
6.2	Simulation	23
6.2.1	Informationsphase	23

6.2.2	Selektionsphase	24
6.2.3	Simulationsphase	30
6.3	Schadensauswirkungen	30
6.4	Besondere Felder	31
6.5	Meister-Menu	32
6.6	Ende der Simulation	36
7	Zusammenspiel der Tools	37
7.1	Allgemein	37
7.2	Objektgeneration	37
7.2.1	Syntax des Objekt-Beschreibungsfiles	38
7.3	Das PlantFlagSet	40
7.3.1	Die Flags	40
7.3.2	Beispiel	43
7.4	Das Allfile	44
7.4.1	Allgemein	44
7.4.2	Syntax	45
7.4.3	Allfile-Beispiel	47
7.5	Verhaltensflags und NPC-Strategien	49
7.5.1	Die Flags	49
7.5.2	NPC-Strategie	50
7.5.3	Beispiele	50
7.6	Charakter-File	52
7.6.1	Eigenschaften und Würfe	52
7.6.2	Syntax	54
7.6.3	Die Daten	55
7.7	Zauber und magische Formeln	70
7.7.1	Anwendung von Zaubern	70
7.7.2	Arten von Zauberei	71
7.7.3	Syntax von Zaubern	73
7.7.4	Beispiel zur Sprucherzeugung	75

8	Anpassung eigener Spielsysteme	77
8.1	Was bleibt, was ändert sich?	77
8.2	Fazit	78
9	Szenarion Multi-User-Interface	79
10	Kontakt	81
11	Bugs & Fehler	82
12	Anhang	83
12.1	Szenarion Zauber	83
12.2	Babyschritte...	85

1 Distribution

Der RFE¹ und der RKS² gehören zum *Szenarion*³ Software Paket und sind **NICHT Shareware, NICHT Public Domain, NOCH Freeware.**

Obwohl die vorliegende Software eingehend getestet wurde, gibt es **keinerlei Garantien** in Bezug auf das *Szenarion*. Das Risiko, die Software dieses Paketes anzuwenden, liegt **gänzlich** beim Anwender.

Der (die) Empfänger(in) des *Szenarions* hat das Recht, das *Szenarion* zu **benutzen**, er (sie) erwirbt jedoch **nicht** das Urheberrecht oder sonstige Rechte an der Software.

Der Author (S.Waschk) ist in **keinem Falle** für Schäden haftbar zu machen, die durch Anwendung oder Änderungen des *Szenarions* entstehen, sowie der daraus resultierenden Folgen.

Nichts aus der Dokumentation oder dem Programmpaket darf **reproduziert** oder außerhalb einer **ein-Maschinen-Umgebung** gebracht werden. Einzige Ausnahmen sind Public Domain Programme, die vom *Szenarion* aufgerufen werden, allerdings nur im Rahmen der vom Urheber gegebenen Richtlinien.

Programm und Dokumentation sind ©**Copyright** geschützt. Alle Rechte vorbehalten.

Findet das *Szenarion* Anklang, so wird es gegebenenfalls "Updates" und Änderungen auf Kundenwunsch geben. Ich werde jedoch nicht für eine Bande von Raubkopierern meine Zeit verschwenden.

Ein riesiges **Dankeschön** an Marc Zimmermann, für die vielen Nachmittage des Assembler-Tunings bei Tee und Keksen und Frank Gerberding, für seine wertvollen Modula II Programme, den Versuchsopfern Hawk, Gronar, Chiage, Huma II., Morgai und Gurth für ihre manigfaltigen Verbesserungsvorschläge, sowie "last but not least" Tina für ihre übermenschliche Geduld in den letzten zwei Jahren.

S.Waschk

¹Rollenspiel-Feld-Editor ©1995

²Rollenspiel-Kampf-Simulator ©1995

³Rollenspiel-Software-Paket ©1995

2 Allgemeine Informationen

2.1 Warum brauche ich das Rollenspiel-Software-Paket?

In meinen langen Jahren als Meister einer Rollenspiel-Gruppe hatte ich schon oft die Idee, mir eine Simulation zu schreiben, die das Nachschlagen in Tabellen abnimmt, die Zeit läßt, die Ideen meiner Spielercharaktere umzusetzen und Monster einigermaßen intelligent handeln zu lassen. Dabei sollte der Meister jederzeit ein **Veto-Recht** haben und in das Handlungsgeschehen eingreifen können. Ein erster Schritt in diese Richtung war mein **MERSsimulator** auf dem C64. Leider konnte er aufgrund von Speichermangel höchstens zwölf Charaktere verwalten, was bei sieben Spielercharakteren einfach zu wenig war.

Anfangs wollte ich das Programm einfach nur auf den Amiga übersetzen, doch lernte ich sehr bald die wirklich einfach zu handhabende Betriebssystem- Routinen lieben.

So kommt es, daß der neue Feld Editor nicht mehr nur 20x20 Kästchen mit verschiedenen ASCII-Zeichen füllt, sondern das geistige Modell zu modellieren hilft, mit dem der Meister seine ihm anvertrauten Charaktere in die Welt der Mystik versetzen kann.

Wer nun sagt, dies kann man auch nur durch geistige Vorstellungskraft erreichen, hat recht, nur ist der Aufwand unverhältnismäßig größer und oftmals, nach zehn Stunden non-stop-Rollenspiel, bleiben die so liebevoll erdachten Szenarien auf der Strecke.

Wenn ein Trupp von fünf finster dreinblickenden, bis zu den Zähnen bewaffneten Orks eine Straße blockiert und den "Helden" zuruft, "**Halt, Gewürrm, oder ihrrr seidd des Toddde!**" , der Magier sich aber erst mal am Kopf kratzt und mit dem Mund voll Chips träge und unwürdevoll magische Worte stammelt, der Zwerg lustlos in der Nase bohrt und der Halbling sich gar schon seit einer Stunde unter dem Tisch zur Ruhe begeben hat, dann ist es höchste Zeit für eine realistischere Version der Fantasy mit fairen Chancen für **jederork!**

2.2 Vollversion

Um die Vorzüge des *Szenarions* nutzen zu können, bedarf es eines intelligenten Multitasking-Betriebssystems. Versuche der PC-Welt, Multitasking “nachzubilden”, sind im Vergleich zum Amiga-Betriebssystem mit einem großen Bedarf (Verschwendung?) an Ressourcen verbunden (Eine bunte Benutzeroberfläche mit vielen lustigen Knöpfchen, die seine willigen Nutzer zu β -Testern und unfreiwilligen Bug-Findern degradiert, kann einem stabiles Betriebssystem einfach nicht das Wasser reichen, und wer will schon noch 95 Jahre auf eine fehlerfreie Windoofs95 Version warten, die jetzt schon bei 16 MB⁴ swapped?)⁵. Damit beschränkt sich die Möglichkeit, das *Szenarion* zu nutzen, vorerst auf den elitären Kreis der Amiga-Gemeinde. Ggf. wird es später einmal eine UNIX-Version unter X-Windows geben... Das Programm wurde auf einem A2000/68030/7MB/532MB-HD unter OS 3.1 erstellt und getestet. Mindestvoraussetzungen für die Vollversion sind:

- Kick 2.0+
- 1MB+ CHIP RAM⁶ und 1MB FAST RAM
- Festplatte mit 5 MB freiem Speicher
- Flicker-Fixer
(oder Sonnenbrille und Augen mit Nachleuchtnetzhaut:)

2.3 Installation

Die Anordnung und die Namen der Unterverzeichnisse dürfen beim Entpacken der Installationsdisketten nicht verändert werden! Danach können die einzelnen Tools

⁴selber gesehen und gelacht!

⁵natürlich gibt es auch bessere Betriebssysteme als OS 3.1, dem Memory-Protection und Resource-Tracking auch noch gut anstehen würde, da aber z.B. LINUX PD und nicht von Microsoft ist, wird es sich kaum gegen die PC-Herdentrieb durchsetzen können...

⁶Die auf Feldern dargestellten Objekte werden je nach Entfernung zum Beobachter berechnet und nach dem LRU-Verfahren im Chip-RAM gepuffert. “Viel” Chip-RAM beschleunigt somit auch die Visualisierung.

im Unterverzeichnis *Szenarion* von einer Shell, CShell (..) aus, die einen **Stack** von mindestens 250000 Bytes aufweisen sollte, gestartet werden.

Für die von den Tools (RKSim oder RFEditor) gewählten Screens und der Pfad für den zu benutzenden ASCII-Editor, sowie der eventueller Ausgabemodus für das Kampfprotokoll können im File *ENV/Szenarion.config* an die eigene Rechnerplattform angepaßt werden. Die einzelnen Schlüsselwörter haben dabei folgende Bedeutung:

- **Path:** Hier ist der Pfad und der benutzte Editor einzutragen (z.B. SYS:Tools/-MEmacs).
- **Screenmode:** Dieser Eintrag bestimmt den zu öffnenden Screen. Folgende Einträge sind zulässig: NTSCHIGHRESLACED (A2000, A3000, A4000 + Flickerfixer), EURO36HIGHRESLACED (A2000, A3000, A4000 + Flickerfixer und mind. ESC-Chipset).
- **Output:** Soll der RKSim ein Kampfprotokoll erstellen, so sind statt dem Eintrag "none" entweder "TeX" oder "ASCII" einzusetzen.

Leerzeichen zwischen Schlüsselwort, Komma und Kommando sind dabei leider unzulässig!

2.4 Input

Um mit zwei bis vier aktiven Spielern einen Kampfschauplatz zu simulieren, ist das *Szenarion*-Multi-User-Interface zwar nicht zwingend erforderlich, es bietet aber mehr Komfort als sechs Hände, die die Tastatur des Meisters mißhandeln. Für einen Spieler kann an Port 1 auch ein normaler Joystick angeschlossen werden.

- Maus
- Tastatur
- *Szenarion*-Multi-User-Interface

3 Warnung

Lesen Sie sich bitte das Kapitel **Bugs** sorgfältig durch, damit Ihre Landschaftschöpfung nicht durch einen Programmfehler zerstört wird.

Es wird darauf aufmerksam gemacht, daß jede Vollversion des *Szenarions* eine kodierte Kundennummer an den verschiedensten Programmstellen enthält. Jede Raubkopie⁷ gefährdet die Weiterentwicklung dieser für den kleinen Kreis der Rollenspieler gedachten Software.

Bei der Shareware-Version wurden bewußt einige Funktionen des RFEeditors entfernt. So können etwa selbst erstellte Landschaften NICHT abgespeichert werden. Bei der Vollversion sind diese Routinen natürlich aktiv und garantieren unbegrenzten Spielspaß.

⁷Sollte mein Magier die magische Aura einer Raubkopie lokalisieren, so habe ich keine Hemmungen, dem Dieb die oben abgebildeten fünf Charaktere oder Schlimmeres zu schicken! Seid gewarnt! (v.r.n.l.): Morgai, Kunstliebhaber aus dem Nebelgebirge, Huma II., Lehrling De Morgans, Gurth Troublemaker Lesemann, Fuzzy-Hawk aus den Schluchten des Affengebirges und Gronar, Betriebswirt aus Moria.

4 Arbeitsablauf

Der ideale Nutzer dieser Software muß nicht über tiefe Programmier-Erfahrung verfügen. Er sollte jedoch mit dem grundsätzlichen Gesamtkonzept seines Rechners vertraut sein, sowie mit einem Text-Editor und der Amiga-Shell bzw. CShell (..) umgehen können.

4.1 Meister-Tools

Folgende Tools muß der Meister besitzen, um mit dem *Szenarion* zu arbeiten:

- *Editor*: Ein vernünftiger Editor, mit dem der Meister Charakter-Daten, das Allfile und das Lib.plt bearbeiten und abspeichern kann (z.B. der Standard-Editor **MEmacs** oder **CygnusEd**).
- *RFEEditor*: Der Rollenspiel - Feld - Editor, mit dem der Meister den Kampfschauplatz erstellen und bearbeiten kann.
- *RKSim*: Der Rollenspiel - Kampf - Simulator. Er erst erweckt den mit dem RFEEditor erstellten Schauplatz und die durch das allfile gewählten Charaktere zu Leben.

4.2 Allgemein

Natürlich kann und soll die Software den Meister nicht ersetzen. Sicher, es ist möglich, zufällig Labyrinth zu konstruieren und seine Spielercharaktere dort "hindurchzuprügeln". Meiner Ansicht nach kann das aber nicht Sinn und Zweck des Rollenspiels sein. Auch verstehe ich das *Szenarion* nicht als "Muß" für ein gelungenes Rollenspiel.

Vielmehr bietet es die Chance, die Qualifikationen der Charaktere neutral zu beurteilen.

Dabei obliegt es der Sorgfaltspflicht des Meisters, die Konfrontationen ausgewogen zu gestalten. Eine Computer-Vollsimulation des Kampfschauplatzes bietet hierfür die optimale Grundlage.

Handlung und Ablauf sind sehr flexibel gestaltbar. Das *Szenarion* ermöglicht mehr, als nur die bloße Simulation eines blutigen Kampfes. Den Spielern stehen, wie im Dialog mit dem Meister, eine Vielzahl von Entscheidungsmöglichkeiten offen. Aber auch darüber hinaus ermöglicht die permanente Eingriffsmöglichkeit des Meisters die Verwirklichung aller Ideen (zumindest der resultierenden Konsequenzen für den Kampf) und somit echtes Rollenspiel.

4.3 Szenario erstellen

Planung des Szenarios

Neben der klassischen Situation, dem Kampf bis zum Tode, kann der Meister den Kampf auch mit Erreichen eines bestimmten Zieles oder der Flucht vom Schauplatz beenden. Für diese Szenarien eignet sich das *Szenarion* besonders gut:

- **Kampf bis zum Tode:** Die Simulation endet mit dem Tod oder Unterwerfung der gegnerischen Partei(en). Bsp: Kampf in einer Arena.
- **Erreichen eines Zieles:** Das Ziel kann, wie beim Kampf bis zum Tode, das Bezwingen der Gegner, das Erreichen eines Punktes auf dem Kampfschauplatz, das Aushalten für eine bestimmte Zeit (..) sein. Bsp: Öffnen eines Tores, durch das dann Charaktere der eigenen Partei zur Hilfe kommen.
- **Flucht:** Orte auf dem Kampffeld müssen erreicht werden, von dem ab ein Charakter als geflohen gilt.

Bevor man mit der Konstruktion des Kampfschauplatzes beginnt (RFEditor), sollte man zu Karopapier und Bleistift greifen und eine Skizze anfertigen. Danach

kann man mit dem RFEeditor (siehe dort) zur Tat schreiten.

RFEeditor

Der RFEeditor dient zur Konstruktion des Kampfschauplatzes und kann von einer Shell oder durch das Anklicken des Icons gestartet werden. Er erzeugt das *name.map*-File und das zugehörige Beschreibungsfile, das *Allfile*.

Siehe auch Rollenspiel-Feld-Editor

Allfile ausfüllen

Ist der Kampfschauplatz erstellt und abgespeichert, so wird automatisch ein "Allfile" erstellt. Dort ist Platz für Informationen, die der Meister den Spielern über örtliche Gegebenheiten und Gegner mitteilen möchte.

Dort entscheidet der Meister auch darüber, wie seine "Monster" sich verhalten sollen. Die Monster und Spieler-Charaktere müssen nun noch beschrieben werden (falls das File noch nicht vorhanden ist).

Siehe auch Allfile

Lib.plt

Um einen möglichst konsistenten Text zu erhalten, existiert eine kleine Wörter-Bibliothek im config-Verzeichnis. Dort sind alle verwendeten Objekte und Charaktere aufgeführt und im Hinblick für die benötigten Formulierungen in der Simulation vordekliniert. Das File muß bei der Erschaffung eines neuen Charakters oder Objektes um die entsprechenden Terme erweitert werden.

Charakter-Files

Zu jedem Spieler- und Nicht-Spieler-Charakter existiert im characters/-Verzeichnis ein Beschreibungsfile mit der Endung *.char*. Bei der Erschaffung eines neuen

Charakters muß auch ein neues File angelegt werden, analog zu den bereits existierenden Charakteren.

Siehe auch Charakter-File

RKSim

Nun kann der Rollenspiel-Kampf-Simulator gestartet werden. Jede Phase der Simulation wird protokolliert und kann in ein File (ins RAM:) umgelenkt (rksim >ram:myFight) oder direkt in der Shell ausgegeben werden.

Siehe auch Rollenspiel-Kampf-Simulator

Nachbereitung

Wird die Simulation beendet, werden die aktuellen Charakter-Daten in das Characters/-Verzeichnis kopiert. Statt der Endung name.char findet man sie unter name.new. Will man die nächste Simulation mit den neuen Werten durchführen, sind die neuen Files über die alten zu kopieren.

4.4 Kurzübersicht

Das folgende Diagramm faßt die vorher beschriebenen Vorgehensweisen zusammen und soll den schrittweisen Arbeitsablauf verdeutlichen, der zum Erstellen eines Kampfschauplatzes notwendig ist:

5 Der Rollenspiel-Feld-Editor

5.1 Grundlagen

Nach dem Start (RFEditor) erscheint ein leeres Feld⁸(empty.map) in maximaler Größe (100x100 Kästchen). Danach sollte man das Feld erst einmal auf die gewünschte Größe einstellen.

Die Eingabe erfolgt mittels eines Cursors. Er wird mit den Pfeiltasten oben rechts gesteuert und kann in der Größe (affect/unaffected) verändert werden. Grundsätzlich werden alle Felder innerhalb des Cursorbereiches von RAISE/PLANT betroffen, die äußeren Randfelder werden lediglich an die landschaftlichen Gegebenheiten angeglichen.

Am besten, man formt zuerst grob die zu erschaffene Landmasse und setzt erst dann Verbindungen zwischen Treppen, stellt Fallen⁹ auf und postiert Charaktere, wenn die Landschaft fertig ist.

Man kann mit **Copy, Rotate und Flip** sehr schnell und leicht aus einer kleinen Fläche eine ganze Landschaft aufbauen.

Die mitgelieferten Landschaften können selbstverständlich auch als Basis neuer dienen oder selbst verändert werden.

Zu dem erstellten Feld wird automatisch ein Allfile erzeugt, in dem die besonders gekennzeichneten Felder genauer definiert werden. **Siehe auch Save, Allfile.**

Bei 68000 Prozessoren ist es empfehlenswert, den angezeigten Feldausschnitt auf 9x9 Felder zu verkleinern.

Siehe auch View.

⁸Das Feld besteht nur aus Wasserfeldern (blau), auf ihnen kann man keine Objekte erzeugen.

⁹Bisher ist das Positionieren von Fallen nicht implementiert. Verbindungen können zwar schon vom Editor erzeugt werden, in der Simulation sind sie bisher aber unwirksam!

5.1.1 Maßstab

Ein Kästchen auf dem Feld entspricht zwei mal zwei Metern, man kann also einen Kampfschauplatz von bis zu 200 mal 200 Metern kreieren.

5.1.2 Die Modi

Es gibt drei verschiedene Modi im RFEeditor:

- **SQUARE:** Die erzeugten Flächen werden mit einem Karomuster gefüllt, in Abhängigkeit zu der gewählten Basisfarbe. Dieser Modus ist zu Beginn aktiv! Durch das Anklicken eines Feldes wird das PlantFlagSet ausgelesen. COLOR ist inaktiv.
- **COLOR:** Die erzeugten Flächen werden mit der aktuellen Basisfarbe gefärbt. Bei Anklicken eines Feldes auf dem Bearbeitungsausschnitt wird dieses ebenfalls mit der aktuellen Farbe gefüllt. SQUARE ist inaktiv.
- **Weder COLOR noch SQUARE:** Sind beide Modi inaktiv, so kann auf die Eckpunkte der Felder zugegriffen werden, indem man diese anklickt. Ein Anklicken von Feldern bewirkt lediglich die Versetzung des Eingabecursors!

Siehe auch SQUARE, COLOR

5.1.3 Objekte

Objekte sind kleine “Bilder”, die mit Plant auf ein Feld gebracht werden können. Ein Feld kann höchstens mit einem Objekt belegt werden. Das aktuelle Objekt wird

in einem kleinen Fenster auf der rechten Seite angezeigt. Wird auf dieses Fenster geklickt, so erscheint ein Fenster mit dem vorgegebenen PlantFlagSet. Die Objekte sind im File data.plt beschrieben und können dort auch verändert werden.

Siehe auch Objektgeneration

5.2 Funktionstasten

Es folgen die Funktionsbeschreibungen der im Fenster des RFEEditors sichtbaren Benutzer-Tasten (Gadgets). Diese können durch einmaliges 'Klicken' mit der linken Maustaste oder teilweise über die Tastatur (Zahlenfeld rechts) aktiviert werden.

Load

Laden eines bereits gespeicherten Feldes (z.B. empty.map zum löschen der gesamten Karte) aus dem Verzeichnis map/. Es erscheint ein Filerequester und listet alle Karten des Verzeichnisses auf.

Save

Speichern eines editierten Feldes ins Verzeichnis map/, das neue Allfile wird automatisch über das alte im Verzeichnis allfile/ gespeichert.

3-D

Feld räumlich darstellen. Dies dauert länger als die 2-D Darstellung und ist nicht so genau, falls man auf einem Feld etwas 'anklicken' will.

2-D

Feld von Oben sehen. Ideal, um Mauern oder Objekte zu setzen. Nachteil: Man sieht keine Änderungen der Höhe von Feldern.

COLOR

Ist dieser Modus gewählt, so werden alle Veränderungen des Feldes in der eingestellten Farbe vorgenommen. Ist weder SQUARE noch COLOR gewählt, so kann

man auf dem Feld direkt Veränderungen vornehmen, indem man auf die Eckpunkte der Felder 'klickt'. Die Ecken werden dann um den eingestellten Höhenfaktor erhöht, wenn man den Mauszeiger nach oben bewegt und dann 'klickt', Äquivalentes gilt für das Erniedrigen.

SQUARE

Hier werden die Veränderungen als Karomuster vorgenommen, als Basisfarbe dient die eingestellte Farbe. Ist weder SQUARE noch COLOR gewählt, so kann man auf dem Feld direkt Veränderungen vornehmen, indem man auf die Eckpunkte der Felder 'klickt'. Die Ecken werden dann um den eingestellten Höhenfaktor erhöht, wenn man den Mauszeiger nach oben bewegt und dann 'klickt', äquivalentes gilt für das Erniedrigen.

RAISE

Erhöhen/Erniedrigen/Level der betroffenen Felder. Es erscheint ein Menu, in dem die gewünschte Funktion ausgewählt werden kann. Random erhöht und erniedrigt zufällig um den angegebenen Höhenfaktor. Der Höhenfaktor kann durch den Balken manipuliert werden.

SIZE

Einstellen der Feldgröße mittels eines kleinen Menu. Die aktuelle Feldgröße wird in der Karte invers dargestellt und kann mittels Menu zwischen 21x21 und 100x100 Feldern variiert werden.

VIEW

Einstellen des angezeigten Bildausschnittes mittels Menu. Die Feldgröße kann zwischen 9x9 und 21x21 Feldern variieren. Die Größe hat enormen Einfluß auf die Darstellungsgeschwindigkeit.

UNDO

Rückgängig machen der letzten Änderung des Feldes. Achtung, **UNDO** ist endgültig und hat erst wieder nach der nächsten Änderung auf dem Feld Wirkung.

COPY

Kopieren eines Bildausschnitts: 'Anklicken' eines Feldes, Copy 'klicken', anderen Randpunkt wählen, Copy 'klicken'. Der ausgeschnittene Bereich wird in einem kleinen Fenster oben links angezeigt. Maximal 21x21 Felder. Durch Drücken der Insert-Taste wird der Inhalt des Copy-Fensters wieder eingesetzt. Dabei dient die aktuelle Cursor-Position als Ursprung für die Ersetzungsfunktion.

ROTATE

Der mit Copy ausgeschnittene Bereich wird rotiert und kann dann mit Insert in das Feld eingesetzt werden (siehe COPY).

FLIP

Spiegelung des mit Copy ausgewählten Bereiches an der Mittelsenkrechten. Mit Insert kann das Feld wieder eingesetzt werden (siehe COPY).

BIND

Erzeugen einer Verbindung zwischen zwei Punkten. So können Keller und verschiedene Stockwerke oder sogar Teleportationspunkte erzeugt werden! Am besten setzt man ein neues Stockwerk so, daß es von außen nicht erreichbar ist. Werden solche Verbindungen in der Simulation benutzt, so sammeln sich die Charaktere um das entsprechende Objekt am anderen Ende der Passage. Also Platz lassen!

SHOW

Anzeigen aller Verbindungen des Feldes durch Linien auf der Karte oben rechts. Mit Break können die Verbindungen gelöscht werden.

BREAK

Löschen einer Verbindung zwischen zwei Punkten, indem man auf die hervorgehobenen Endpunkte auf der Karte oben rechts 'klickt'. Es werden immer beide Verbindungspunkte gelöscht.

AFFECT / Taste +

Cursorbereich vergrößern. Mögliche Cursorgrößen sind 1x1, 3x3, 5x5, 7x7 und 9x9 Felder. Dieser Bereich kann dann je nach aktivem Modus (z.B. SQUARE) bearbeitet werden.

UNAFFECT / Taste -

Cursorbereich verkleinern. Mögliche Cursorgrößen sind 1x1, 3x3, 5x5, 7x7 und 9x9 Felder. Dieser Bereich kann dann je nach aktivem Modus (z.B. SQUARE) bearbeitet werden.

Pfeiltasten / Zahlenfeld-Tasten 8,6,2,4

Mit den Pfeiltasten oben rechts auf dem Window kann der Eingabecursor auf dem Feld positioniert werden.

NEXT / Taste]

Für Plant Objekt auswählen (Nächstes in der Liste). Objekte sind mit bestimmten Flags behaftet (siehe PlantFlagSet). Die Flags können durch "anklicken" des aktuell eingestellten Objektes ausgelesen werden. Eine Veränderung ist hierbei allerdings nicht möglich.

PREVIOUS / Taste [

Für Plant Objekt auswählen (Vorgänger in der Liste). Siehe auch NEXT.

Taste /

Das erste Plant Objekt aus der durch das data.plt-File bestimmten Liste wird ausgewählt.

Taste *

Das letzte Plant Objekt aus der durch das data.plt-File bestimmten Liste wird ausgewählt.

PLANT / Zahlenfeld-Taste 5

Aktuelles Objekt auf alle Felder setzen, auf denen der Eingabecursor aktiv ist. Objekt empty wird auf dem Feld nicht angezeigt (ignoriert). Objekte vererben die mit ihnen verbundenen Flags an das Feld, auf das sie plaziert werden. Zum Löschen eines Objektes muß das Objekt “empty” (erstes Objekt) gesetzt werden. Siehe auch NEXT.

Objekte mit den gesetzten PlantFlags *Story* oder *Charakter* findet man später unter Koordinaten- und Namensangabe im Allfile (*siehe dort*) wieder.

5.3 Grafische Fenster-Oberfläche

Farbpalette

Je nach aktivem Modus hat man die Auswahl zwischen 13 bzw. 15 Farben. Die Farben blau und dunkelrot können nur im COLOR-Modus ausgewählt werden. Blaue Felder werden in der Simulation als Wasserfelder betrachtet und behindern die Bewegung oder machen sie ganz unmöglich. Dunkelrote Felder repräsentieren Mauern. Mauern werden erst ab zwei zusammenhängenden Feldern dargestellt. Eine zusätzliche Manipulation der Feldeigenschaften durch das PlantFlagSet ist jederzeit möglich (z.B Weg Sperren).

- *Blau*: behindernde Wasserfelder
- *Rot*: Mauern
- *Grau*: begehbare Felder

Karte

Die Karte oben rechts zeigt die gesamte Landschaft im Überblick und den aktuellen Arbeitsbereich. Durch “anklicken” eines bestimmten Punktes auf der Karte kann der Bearbeitungsausschnitt neu gewählt werden. Zwei Positionierungsbalken referenzieren den aktuell dargestellten Ausschnitt.

Bearbeitungsausschnitt

Der Arbeitsbereich dient zum Editieren der Landschaft. Je nach Modi können verschiedene Aktionen ausgeführt werden:

- *Erhöhen/Erniedrigen*: Alle Felder, auch Wasser und Mauern, können erhöht bzw. erniedrigt werden, unabhängig vom aktiven Modus.
- *Färben*: Im Modus COLOR können die vom Cursor betroffenen Felder in der aktuellen Farbe eingefärbt werden.
- *Karo-Muster*: Im Modus SQUARE können die vom Cursor betroffenen Felder in einem von der aktuellen Farbe abhängigen Muster gefärbt werden.
- *Mauern*: Im Modus COLOR können Felder mit der dunkelroten Mauerfarbe gefärbt werden. Ab zwei zusammenhängenden Feldern wird die Mauer dargestellt.
- *Objekte*: Durch PLANT kann der aktuelle Cursor-Bereich mit dem aktuell eingestellten Objekt belegt werden.

5.4 Tips zur Felderzeugung

Anwender: “ Wie zum Henker kann ich...”

“... überhaupt Land erzeugen”

Antwort: “Mit den Cursortasten kann der Eingabecursor auf die gewünschte Stelle auf der Karte gebracht werden, dort kann dann mit RAISE Land erzeugt werden!”

“...die Ecke eines Feldes erhöhen?”

Antwort: “Benutze Raise und ändere in dem Menu den Balken!”

“...das Copy Fenster schließen?”

Antwort: “einfach auf das Fenster, nicht aber auf INSERT drücken !”

“...die FeldFlags auslesen?”

Antwort: “ Unter SQUARE einfach auf das betroffene Feld 'klicken'.”

“...die FeldFlags ändern?”

Antwort: “FeldFlag auslesen und gewünschte Flags 'an-' oder 'ausklicken'.”

“...Mauern erzeugen?”

Antwort: “Mit COLOR Feld in der 2.Farbe der Farbpalette färben. Mauern werden erst ab zwei zusammenhängenden Feldern dargestellt.”

“...Objekte löschen?”

Antwort: “Es existiert ein Objekt “empty”. Wird es über ein existierendes Objekt gespeichert, so wird kein Objekt auf den betroffenen Feldern mehr dargestellt.”

“...Felder löschen?”

Antwort: “Entweder Höhenfaktor auf Null und mit RAISE → level und Farbe blau füllen oder einen leeren Ausschnitt mit COPY über die betroffene Stelle setzen.”

“...Felder unpassierbar machen?”

Antwort: “Entweder durch das Setzen einer Mauer oder durch das Plantflag Block-Way.”

Es empfiehlt sich, die mitgelieferten Szenarios, die im Verzeichnis maps/ und all-file/ näher beschrieben sind, heranzuziehen und als Vorlage für eigene Simulationen zu benutzen.

6 Rollenspiel-Kampf-Simulator

6.1 Arbeitsweise

Der RKSIm dient zur Simulation der mit dem RFEditor erstellten Kampfschauplätze. Er liest zu Beginn die Felddaten (.map) und das vom Meister erstellte Beschreibungsfile (.alf) ein (*siehe auch Allfile*). Danach werden, entsprechend der Angaben des Allfiles, die beteiligten Charaktere geladen.

Sind alle Datenfiles fehlerfrei, so startet die Simulation. Je nach Anzahl der Spielercharaktere erscheinen ein bis vier unabhängige Fenster auf dem Bildschirm.

Dem Meister kommt hierbei nur eine überwachende Rolle zu. Ist das Szenario ausgewogen gestaltet und ausreichend getestet worden, so kann er sich getrost auf das weitere Rollenspiel vorbereiten und die Charaktere ihrem "gerechten" Schicksal überlassen.

Nichts desto trotz kann der Meister jederzeit über "Hot-Keys" eingreifen und über ein Pop-Up-Menü das aktuelle Geschehen bestimmen. Sollten ihre Charaktere ausgefallener Ideen haben, als die bisher vorgesehenen¹⁰, so können zumindest die Folgen für den Kampf so verwirklicht werden.

¹⁰Ist in der von mir gemasterten Gruppe "seltenst" der Fall, sollten sich jedoch viele Nutzer des *Szenarions* für bestimmte weitere Menü-Punkte einsetzen, werde ich diese ggf. implementieren oder darüber nachdenken, das Menü dynamisch zu gestalten.

6.1.1 Die Charakterfenster

Dieses Fenster bildet die Schnittstelle zwischen jedem Spieler-Charakter und dem Kampfschauplatz. Die Daten haben folgende Bedeutung:

- **Abbildung des Sichtfeldes:** Zentral im Charakter-Fenster angeordnet ist der sichtbare Bereich des Kampfschauplatzes (das, was der Spielercharakter von seiner Position aus sehen kann). Dabei können Erhöhungen dahinter liegende Bereiche verdecken oder die Wahrnehmung des Charakters mindern. Mauern beschränken ebenfalls die Sicht, mit Ausnahme von Mauerelementen, die ein Fenster beinhalten. Je weiter ein Objekt entfernt ist, desto schwerer ist es wahrzunehmen. Der Spielercharakter selbst befindet sich in der Mitte des Feldes.
- **Name und Abbild** des Charakters in der oberen linken Ecke ordnen das Fenster dem jeweiligen Spielercharakter zu. Das rechts davon angeordnete Balkendiagramm zeigt die dem Charakter verbleibende Lebensenergie an.
- **Freund-Ikonen:** Bis zu neun befreundete Charaktere werden samt Kurzinfo dargestellt. Sie befinden sich, falls vom Meister Nichtspieler-Charaktere als *DisplayNPCharacter* eingestellt wurden, in der oberen rechten Ecke. Zusätzlich zeigt ein in der Ikone rechts angeordnetes Balkendiagramm¹¹ deren physische Verfassung und ein Bild¹² in der oberen linken Ecke die Entfernung des ikonifizierten Charakters zum Spielercharakter.

¹¹der Balken kann blau oder dunkelgrau dargestellt sein. Blau bedeutet dabei, der Charakter kann frei handeln, grau, der Charakter ist bewußtlos oder stirbt.

¹²Ein schematisierter Mensch, ein Auge oder kein Bild. Der Mensch bedeutet, der Charakter kann berührt werden (Abstand ca. zwei Meter), das Auge zeigt an, der ikonifizierte Charakter ist in unmittelbarer Umgebung. Wird kein Bild dargestellt, so befindet sich der Charakter mehr als 10 Meter entfernt.

- **Text-Segment:** Die untere Hälfte des Charakter-Fensters dient zur Ausgabe von Spielerinformationen und zur Menu-Auswahl mittels Multi-User-Interface.

Die Anzahl der erscheinenden Charakter-Fenster ist abhängig von der Zahl der Spieler-Charaktere, die im Allfile (*siehe dort*) referenziert wurden. Es können bis zu vier Spieler-Charaktere eingestellt werden, die ihr jeweils eigenes Fenster und Eingabe-Interface (Multi-User-Interface) erhalten. Weitere Spieler-Charaktere können über die Verhaltensflags (*siehe dort*) Handlungsrichtlinien an ihre Charaktere geben oder sich einem Spieler “anvertrauen”.

6.2 Simulation

Die Simulation des Kampfschauplatzes kann auf zwei verschiedene Arten erfolgen. Zum einen kann die reale Gruppe des Meisters den Kampfschauplatz durchspielen, oder, der Meister führt eine Vollsimulation aus. Dazu stellt der Meister im Meistermenu mittels der **Simulate x Rds.-**Taste die Anzahl der auszuführenden Runden ein (*siehe dort*).

6.2.1 Informationsphase

Zu Beginn jeder Runde wird die Umgebung eines jeden Charakters berechnet. Daraus ergeben sich folgende Informationen: Wen sieht der Charakter, wie gut kann er das Umfeld einschätzen und nicht zuletzt, wen kann er angreifen. Es kann vorkommen, daß der Spielercharakter von Gegnern beschossen oder “bezaubert” wird, die er, aufgrund eines ungünstigen Standortes, nicht sehen kann und die ihn unvorbereitet treffen.

Ist die Analyse des Umfeldes abgeschlossen, so werden die ermittelten Daten als Sichtfeld und als Text visualisiert.

Dabei entscheidet jeder Spieler per Knopfdruck, wann er neue Textinformationen erhalten möchte, bis der ganze, für seinen Charakter erzeugte Text angezeigt wurde. Nur der Meister kann mittels Tastendruck (**!alt s**) eine Unterbrechung der Informationssequenz erzwingen.

6.2.2 Selektionsphase

Der folgende Graph zeigt die im RKSim vorgesehenen Standardhandlungen, die Spieler- und Nicht-Spieler-Charakteren zur Verfügung stehen.

In dieser Phase kann der Spieler seinen Charakter zu Handlungen bewegen, die durch die Menge der erhaltenen Informationen beschränkt sind. So können z.B. Feinde selbst dann nicht angegriffen werden, wenn sie unmittelbar beim Spielercharakter stehen und ein modifizierter Wahrnehmungswurf mißlungen ist (Der Charakter ist dann verwirrt und kann sich für keinen Gegner entscheiden). Dies ist jedoch eher eine Ausnahme!

Mittels des Multi-User-Interfaces können alle Spieler gleichzeitig die Handlung ihres Charakters bestimmen. Ein kurzer Druck auf den Knopf des Interfaces bewirkt ein Springen zum nächsten Menu-Punkt (ein Pfeil erscheint in der unteren rechten Ecke und zeigt auf den nächsten dunkel dargestellten Menu-Punkt), ein längerer Druck dient zur Selektion des aktuell angezeigten (hellen) Menu-Punktes, ein rot ausgefüllter Kreis erscheint anstelle des Pfeiles. Wird der Druckknopf des Interfaces darüber hinaus betätigt, so erscheint das *back*-Zeichen, mit dem man zurück zur

Wurzel des Menu gelangt (Aktion **Abwarten**).

Falls **kein Multi-User-Interface** zur Verfügung steht, können die Spieler auch über die Tastatur ihre Charaktere steuern:

Dazu stehen jedem Spieler zwei Tasten zur Verfügung, eine zum Durchlaufen des Menu, eine weitere zur Selektion:

Spieler	Fenster	nächster Menu-Punkt	Selektion / more
1	oben links	Q	A
2	oben rechts	I	K
3	unten links	7	4
4	unten rechts	9	6

Diese Belegung der Tasten sollte verhindern, daß sich die Spieler während eines Kampfes die Arme verknoten. Für den häufigen Gebrauch kann ich das Multi-User-Interface nur wärmstens empfehlen.

Zur Auswahl steht ein nach Ebenen und Oberbegriffen geordnetes Menu (*siehe Graph*) bereit. Mittels Selektion eines Oberbegriffes erscheinen neue Unterbegriffe, die eine Vielzahl von Entscheidungsmöglichkeiten bieten. Ich verwende im weiteren die üblichen Begriffe PC (Player Character = Spieler-Charakter) und NPC (Nicht-Spieler-Charakter).

Hier nun die Menu-Punkte im einzelnen:

- **Abwarten:** Der Charakter verhält sich abwartend. Es sind keine weiteren Unterpunkte vorhanden.

Abwartende Charaktere haben die Möglichkeit, von ihrem Standpunkt aus am Geschehen passiv teilzunehmen. Es empfiehlt sich in der Regel jedoch nicht, sich so zu verhalten (wer nichts macht, macht auch nichts falsch, aber auch nichts richtig). Nicht-Spieler-Feinde nutzen in der Regel jede Schwäche ihrer Gegner aus! In manchen Szenarios entscheiden gerade die ersten Runden über Sieg oder Niederlage einer Partei.

- **Statisches Manöver:** Sonderaktionen.

Hier sind alle Aktionen aufgeführt, die sich nicht unter die Oberbegriffe Kampf und Bewegung einordnen lassen.

Selbst (ansehen) dient zum Anzeigen der eigenen Ausrüstungsgegenstände und der Verfassung. Rüstungen und Rüstungsteile, Schild und griffbereite Waffen werden ausgegeben.

Durch die Funktion **Sieh an** können gezielte Informationen über Spieler- und Nicht-Spieler-Charaktere eingeholt werden.

Mit **nimm** (Fern-/Nahkampfwaffe) wechselt ein Spieler-Charakter seine griffbereiten Waffen. Die neue Waffe kann aus dem Charakter-File (*Schlüsselwort Waffen*) ausgewählt werden. Dabei werden entweder nur Fernkampf- oder Nahkampfwaffen angezeigt.

Die folgenden drei Aktionen finden Sie im Untermenü **Befehl: Befehl/ Sende Getreue** schickt die eigenen Gefolgsleute (mit gesetztem Verhaltensflag *obeye*), die sich in Sichtweite befinden, zu einem ausgewählten Punkt. Hierbei kann es sich um Gegenstände, Gegner, Freunde, Felder mit aktiven Zaubern oder Treppen handeln. Letztere werden benutzt, d.h., Charaktere gelangen an das andere Ende der Passage. Werden die Getreuen in eine Himmelsrichtung geschickt, so laufen sie, anderenfalls gehen sie zu ihrem Ziel. Getreue bestimmt der Meister oder können durch eine flammende Ansprache gewonnen werden (siehe **Rede mit**).

Um das Verhalten von Gefolgsleuten zu bestimmen, kann man sich **Befehl/ haltet ein!** und **Befehl/ greift ein!** bedienen. Wie der Sende-Befehl können diese Befehle nur Charakteren gegeben werden, bei denen das Verhaltensflag *obeyesetzt* ist (s.o.). Bei dem Aufruf zum Eingreifen nehmen passive Charaktere den Kampf gegen alle Gruppenfremde auf. Andererseits stellen sie bei der Negation die Kampfhandlungen ein und verhalten sich passiv.

Will man mit Charakteren sprechen, so kann ein Spieler die Funktion **Rede mit** verwenden. Dadurch erhält man, je nach Charakter-Wert *Auftreten* des Fragenden und *Verhaltensflags* des Gefragten, Informationen. Ein wichtige Anwendung dieser Aktion ist die Aufnahme von Überläufern oder das Gewin-

nen von Einfluß auf Charaktere der eigenen Gruppe. Die Reaktion des Angesprochenen kann von Ablehnung bis fanatischer Bewunderung reichen, die dem Fragenden einen treuen Kampfgefährten beschert. *Siehe auch Verhaltensflags* Ist ein Charakter abgeschnitten, so kann er durch **ruf Hilfe** Freunde zu sich ordern, sofern sie nicht tief in einen Kampf verwickelt sind. Selbst Spieler-Charaktere hören diesen Ruf, der auch von Nicht-Spieler-Charakteren stammen kann, die starke Verletzungen erhalten haben! Es besteht eine gewisse Wahrscheinlichkeit, daß solche Nicht-Spieler-Charaktere sich (bald) ergeben.

Die beschriebenen Aktionen im Überblick

Aktion	Wirkt auf	Auswirkungen	Beschreibung
selbst	PC	weitere Aktion möglich	Ausrüstung und Physe ausgeben
sieh an	NPC's, PC's	erhöhte Wahrnehmung	andere Charaktere ansehen
nimm	Waffen	weitere Aktion möglich	Fern- oder Nahkampfwaffe wechseln
sende Ü.	NPC's	gutes Abwehrverhalten, steht	Untergebene gehen zu einem Ziel
greif ein	NPC's	gutes Abwehrverhalten, steht	Untergebene nehmen den Kampf auf
halte ein	NPC's	gutes Abwehrverhalten, steht	Untergebene stellen Kampfhandl. ein
rede mit	NPC's, PC's	kaum Abwehrbereit, Probe nötig	Reden, Gegner können bekehrt werden
ruf Hilfe	befr.NPC's	gutes Abwehrverhalten	Ruft eigene NPC's zu Hilfe

- **Bewegung:** Erreichen eines Zieles oder Bewegung in eine Richtung. Es liegt auf der Hand, daß unterschiedliche Bewegungsarten verschieden hohe Risiken bergen. Um auf Wasserfelder zu gelangen (schwimmen), muß das Zielfeld sich in einem Gewässers befinden. Schwimmer allerdings laufen Gefahr, zu ertrinken!

Gehen, laufen, annähern und schleichen: Als Bemessungsgrundlage für die Entfernung, die ein Charakter zurücklegen kann, dienen die Charakter-Werte *Geschwindigkeit* und *B&M* (allgemeine Bewegungsfähigkeit). Der Spieler muß lediglich sein Ziel, einen Gegenstand (Ggst.), eine Himmelrichtung (NSOW) oder einen Charakter bestimmen, den er erreichen will. Der Simulator wählt daraufhin den optimalen Weg unter Berücksichtigung der Bewegungen anderer Charaktere. Etwas aus dem Rahmen fällt die Aktion **Verstecken**, mit der ein Charakter für andere Charaktere kaum noch wahrzunehmen ist (Probe auf *verstecken* nötig).

Die beschriebenen Aktionen im Überblick

Aktion	Ziel	Auswirkungen	Beschreibung
gehen	NSOW, Ggst., Char.	keine	normale Bewegung zum Ziel
laufen	NSOW, Ggst., Char.	schlechte Wahrn., leichtes Ziel	läuft zu einem Ziel
annähern	NSOW, Ggst., Char.	besserer Ang.+ Wahrn., sehr langsam	geht bis zu 2 Meter weit
schleichen	NSOW, Ggst., Char.	guter Ang., schlecht wahrzun., langsam, Probe nötig	schleicht maximal 4 Meter
verstecken	PC	opt. Ang.+Wahrn., kaum wahrzun., steht, Probe nötig	auf Feld verstecken

- **Nahkampf** :Angriff eines oder mehrerer Feinde. Vorteilhaft wirkt sich hierbei eine Attacke aus einem höher gelegenen Feld aus (umgekehrt wird ein Angriff aus einer Senke gewertet). Der **Angriff**, **Rundumschlag** und das **Abdrängen** dienen dazu, dem Gegner Wunden zuzufügen. Bei den beiden erstgenannten Aktionen spielt der Charakter-*Nahkampfwaffenwert* die entscheidene Rolle und soll primär Schaden zufügen. Das Abdrängen hingegen erfordert *Stärke* und bietet einem eingekesselten Charakter die Möglichkeit, auszubrechen. Kann der erfolgreich abgedrängte Charakter nicht ausweichen, so erleidet er sogar einen kritischen Treffer. Zum **Rundumschlag**¹³ bleibt noch zu erwähnen, daß jeder Charakter, ob Freund oder Feind, betroffen wird, der auf einem angrenzenden Feld des Angreifers steht. Der Angriff macht jedoch nur einen Bruchteil eines normalen **Angriffes** aus und ist somit nur Kämpfern zu empfehlen, die ihre Waffe sicher beherrschen.

Es empfiehlt sich weiterhin, bei vielen Angreifern Schläge **abzuwehren**. Auch ohne diese Aktion werden Schläge, Geschosse etc. pariert, jedoch nur im Rahmen des eigenen *Defensivwertes*. Durch **Abwehren** erhöht sich der Defensivbonus, sinkt also erst nach einigen Angriffen mehr in den kritischen Bereich.

Die beschriebenen Aktionen im Überblick

Aktion	Gegner	Auswirkungen	Beschreibung
angriff	feindl. Char.	keine	Gegner mit Waffe angreifen
abwehren	PC	Offensivwert zu Defensivw. addieren	setzt alles auf Abwehr
Rundumschlag	alle Umstehenden	greift bis zu acht Gegner an, schlechtere Attacke	nahe Gegner mit Waffe angreifen
abdrängen	feindl. Char.	Stärke nötig statt Offensivwert.	Gegner abdrängen

- **Fernkampf**: Einsatz einer Fernwaffe auf einen Gegner oder zielen. Wie im Nahkampf bestimmt der Kampfwert (hier *Fernkampfwaffen*-Wert) den Erfolg der Aktion. Charaktere, die ihre Fernkampfwaffe nicht sicher beherrschen, sollten vor dem Schuß besser **zielen**. Dies erhöht die Trefferwahrscheinlichkeit

¹³Original-Idee von B. Szabo

des nächsten Schusses. Jede Runde zu feuern, erfordert ohnehin elbenhafte Gewandheit!

Während man sich auf einen Schuß/Wurf vorbereitet oder feuert, ist man gegen feindliche Angriffe kaum geschützt.

Manche Waffen können nur geworfen werden (z.B. Dolch). Sie sind somit nur einmal einsetzbar. Danach kann man mit "nimm Fernkampf-Waffe" eine neue Waffe wählen.

Die beschriebenen Aktionen im Überblick

Aktion	Gegner	Auswirkungen	Beschreibung
zielen	PC	nächster Schuß mit Bonus, kaum Defens.+Wahrn.	auf Gegner zielen
feuern auf	feindl. Char.	kaum Verteidigung.+Wahrn.	auf Gegner feuern

- **Zaubern:** Der Charakter will zaubern oder sich auf einen Zauber vorbereiten. Der elementare Charakter-Wert ist hier der für *Sprüche*. Der Charakter kann aus seinen Spruchlisten einen auswählen und anwenden (**zaubern**), sofern seine magische Aura (*Magiepunkte*) für den Spruch ausreicht.

Wie beim Zielen mit einer Fernkampf-Waffe erhöht das **Konzentrieren** die Erfolgsaussichten eines Zaubers.

Werden diese Aktionen gewählt, so können gegnerische Angriffe nur schwer abgewehrt werden. Unter Umständen hat man sich so stark auf die Magie konzentriert, daß der Gegner, für den der Zauber bestimmt war, nicht mehr wahrgenommen werden kann.

Die beschriebenen Aktionen im Überblick

Aktion	Ziel	Auswirkungen	Beschreibung
konzentrieren	PC	folg. Zauber mit Bonus, kaum Defens.+Wahrn.	Spruch vorber.
zaubern	alle Char./Ggst.	kaum Verteid.+Wahrn.	Spruch anw.

Jede Menu-Ebene enthält eine **zurück** - Funktion, mit deren Hilfe man die jeweils vorhergehende Ebene erreichen kann.

Die Selektionsphase ist zeitlich auf ca. eine Minute begrenzt. Der Meister kann jedoch schon vorher durch drücken von **alt s** die Phase abbrechen oder durch **alt q** die gesamte Simulation beenden.

Die Zeit ist dabei bewußt knapp bemessen, damit der Magier sich nicht aus einem

Wust von Pergamenten vergessene Zauberformeln heraussucht, der Krieger in seinem Sammelsurium an Waffen gemächlich eine passende wählt oder der Halbling nicht lange philosophiert, ob er sich im eigenen Schatten versteckt, wegrennt oder gegebenenfalls wieder einmal gar nichts unternimmt!

Es sollen sich ruhig ein paar Schweißperlen auf der Stirn sammeln...

6.2.3 Simulationsphase

Während dieser Phase werden die vorher in der Selektion getroffenen Entscheidungen umgesetzt. Dabei erhalten alle Spielercharaktere, wie in der Informationsphase, die sie betreffenden Simulationsdaten in ihrem Charakter-Fenster. Es gilt die folgende Handlungsreihenfolge:

1. **Sonderaktion** : Informationsausgabe zu Ansehen, Reden mit und ruf Hilfe. So greift ein "bekehrter" Feind nicht mehr an.
2. **Kampf** : Anhand der Bewegungswerte/Mali/Boni der einzelnen Charaktere wird die Reihenfolge festgelegt, in der die Charaktere angreifen. Ein ausgeschalteter Gegner greift nicht mehr an, falls er erst nach dem erhaltenen Treffer an der Reihe wäre.
3. **Bewegung** : Nach dem Kampf bewegen sich die Charaktere. Es bewegen sich aber nicht nur Charaktere, die dies vorher als Aktion festgelegt haben. Ein Angreifer, der seinen Gegner abgedrängt hatte, setzt nach, einer, dessen Gegner flieht, verfolgt diesen.

6.3 Schadensauswirkungen

Die Auswirkung von Angriffen ist unterschiedlich. Geringerer Schaden führt nur zum Verlust einiger Lebenspunkte. Kritische Treffer jedoch können den hohen Verlust an Lebenspunkten zur Folge haben, Körper- oder Rüstungsteile zerstören, den Verwundeten für einige Runden unbeholfen und/oder benommen machen oder sogar sofort tödlich wirken.

Ist ein Charakter z.B. benommen, so kann er für diesen Zeitraum keine Aktionen ausführen. Er liegt dann benommen auf dem Boden, und muß unter Umständen mit ansehen, wie seine Feinde auf ihn einstechen. NPC's kämpfen in der Regel jedoch nicht gegen hilflose Feinde.

6.4 Besondere Felder

Durch Handlungen der Charaktere (z.B. Anwendung eines Zaubers, der Felder mit Feuer belegt) oder Tod eines Charakters (Leiche bleibt zurück) werden die Grundeinträge der Map geändert. Im *Szenarion* unterscheiden wir permanente und temporäre Veränderungen:

- Charaktere, die sterben, werden aus der aktiven Charakterliste ausgekettet und bleiben als Leiche auf dem Kampffeld bis zum Ende der Simulation liegen. Solche Felder behindern die Bewegung anderer Charaktere nicht und können dem Meister zur Auswertung und Erkennung von Brennpunkten dienen. Eines der rechts abgebildeten Objekte erscheint auf dem Feld.
- Ist ein Charakter durch Trefferauswirkungen kampfunfähig (benommen), so wird er durch eines der rechts abgebildeten Objekte dargestellt. Sobald er wieder zur Besinnung kommt, erscheint er wieder in seiner gewohnten Form.
- Einige Charaktere, die ihren Anführer verloren haben oder durch schwere Treffer geschwächt sind, können sich, je nach Eigenschaftswert ergeben. Diese Charaktere bewegen sich nicht mehr und werden durch das rechts dargestellte Objekt auf dem Kampffeld repräsentiert.
- Wirkt auf einem Feld ein Zauber, dessen Aura Hitzeschaden (Wert,6) verursacht, so erscheinen kleine Flammen (siehe links oben) auf den betroffenen Feldern. Kältesprüche (links unten) werden durch Schneeflocken und kleine Schneeberge (7), Verhüllungszauber wie Nebel (rechts oben) durch Wolken und alle übrigen durch helle Sterne (rechts unten) dargestellt.

6.5 Meister-Menu

Will der Meister in das Geschehen eingreifen, so kann er mittels Tastendruck (**alt shift m**) das Meister-Menu “bestellen” (MASTER REQUEST ACKNOWLEDGED wird angezeigt) Durch nochmaliges Drücken der o.g. Tastenkombination kann der Request zurückgenommen werden. Das Fenster erscheint am Ende der Simulationsphase zentral im Fenster.

Das Fenster enthält eine Reihe von Tasten, mit denen Kampfauswirkungen beeinflusst, neue Charaktere geladen, alte entfernt, Feldflags geändert, oder sogar Zauber auf Feldern (de)aktiviert werden können.

Auch die Kampfprotokoll-Art (TeX/ASCII/kein Output) und der Editor-Pfad können neu eingestellt werden.

Je nachdem, ob auf dem mit dem Meister-Menu bearbeiteten Feld bereits ein Charakter anwesend ist oder nicht, sind verschiedene Funktionstasten aktiv, die mit dem Mauszeiger angewählt werden können:

Pick Field

Eine kleine Karte des Kampfschauplatzes erscheint, auf der mittels Mauszeigers ein Feld zur weiteren bearbeitung ausgewählt werden kann. Zu Beginn ist das Feld des ersten Charakters des Allfiles editierbar. Felder, die einen Charakter oder besondere Gegenstände enthalten, sind auf der Karte dunkler gefärbt. Bei der Selektion eines solchen Feldes wird die Besonderheit vergrößert.

Remove Char

Der auf dem aktuell gewählten Feld (mittels “Pick Field”) anwesende Charakter wird aus der Simulation entfernt. Der Meister kann so z.B. geflohene Charaktere

entfernen.

Ist kein Charakter auf dem Feld präsent, so bleibt diese Taste deaktiviert.

Add Char

Auf das aktuell ausgewählte Feld kann ein Charakter aus dem *characters/* Verzeichnis in das Szenario eingefügt werden (z.B. ein durch einen Magier beschworener Geist).

Ist auf dem aktuellen Feld bereits ein Charakter präsent, so bleibt diese Taste deaktiviert.

LE full

Der auf dem aktuell ausgewählten Feld präsente Charakter erhält seine durch die Körperentwicklung bestimmte maximale Lebensenergie zurück. Will man hier einen anderen Wert eintragen, so kann man die durch "Edit Values" auf dem Charakterbogen realisieren.

Ist auf dem aktuellen Feld kein Charakter präsent, so bleibt diese Taste deaktiviert.

Add Wounds

Der Meister kann hier Charakteren Schaden zufügen, der z.B. durch Sonderaktionen von Spielern entstanden ist. Mittels Schieberegler kann er die Dauer einer eventuellen Benommenheit oder die Runden bis zum Tode einstellen. Zur Kontrolle erscheinen im Feld, in dem der aktuelle Charakter angezeigt wird, die entsprechenden Symbole (Totenschädel = stirbt, liegende Gestalt = Benommen).

Ist auf dem aktuellen Feld kein Charakter präsent, so bleibt diese Taste deaktiviert.

Clear Wounds

Schaden, jedoch nicht der Verlust von Lebenspunkten, wird zurückgenommen. Die im Charakter-Feld angezeigten Symbole (Totenschädel = stirbt, liegende Ge-

stalt = Benommen) verschwinden.

Ist auf dem aktuellen Feld kein Charakter präsent, so bleibt diese Taste deaktiviert.

Player/Non-Player/Show NPC

Dieses Cycle-Feld gibt Auskunft über die Art des auf dem Feld anwesenden Charakters. Mittels “klicken” auf das Wechsel-Symbol kann die Charakter-Art geändert werden. Player-Charaktere müssen von einem realen Spieler gesteuert werden, Non-Player übernimmt der Computer, ebenso Show NPC, diese werden jedoch als Ikone befreundeten Charakteren angezeigt. Zu Beginn sind hier die Einträge aus dem Allfile gültig.

Ist auf dem aktuellen Feld kein Charakter präsent, so bleibt diese Taste deaktiviert.

Acting

Hier kann das Verhalten von Nichtspieler-Charakteren durch Setzen bestimmter Kombinationen von Einträgen aus den Verhaltensflags beeinflusst werden. Ist auf dem aktuellen Feld kein Charakter präsent, so bleibt diese Taste deaktiviert.

Siehe auch Verhaltensflags und NPC-Strategien

Edit Values

Will der Meister Änderungen am aktuell gewählten Charakter vornehmen, die durch die übrigen Funktionen nicht abgedeckt sind, so kann er mit dem im *ENV/-Szenarion.config* eingestellten Editor direkt auf die Charakter-Daten zugreifen (z.B. bei der Eingabe einer neuen Lebensenergie...).

Nach dem Editieren müssen die Daten abgespeichert und die Oberfläche des Szenarions (Screen) durch einen Mausklick aktiviert werden. Erst dann werden die Daten vom RKSIm neu eingelesen.

Ist auf dem aktuellen Feld kein Charakter präsent, so bleibt diese Taste deaktiviert.

Change Flags

Der Meister kann so direkt auf die Plantflags des aktuellen Feldes (wie im RFE-ditor) zugreifen. So können z.B. Felder nachträglich gesperrt (durch BlockView) oder passierbar gemacht werden.

Siehe auch PlantFlagSet

Add Spell/Remove Spell

Zaubersprüche aus dem *spells/* Verzeichnis können so (de)aktiviert werden. Das aktuelle Feld ist dabei als Zentrum der Magie anzusehen. Der Meister kann die Stärke des Spruches vorgeben. Nicht alle Sprüche (z.B. Teleport) zeigen so Wirkung. Sind mehrere Sprüche präsent, so muß "Remove Spell" mehrmals betätigt werden. Die Zauber werden dabei in chronologischer Reihenfolge des Auftretens entfernt.

TeX/ASCII/no output

Dieses Cycle-Feld zeigt an, welche Protokoll-Art im *ENV/Szenarion.config* gewählt wurde. Sie kann mittels des Wechsel-Symbols geändert werden.

Path&Editor

Der durch das *ENV/Szenarion.config* eingestellte Editor (z.B. MEmacs oder CED ...) kann mit dieser Funktion geändert werden. Dazu erscheint ein kleines Fenster

Statistic

Eine Statistik über die an den letzten 20 Kampfrunden beteiligten Charaktere wird erstellt. Dabei werden die einzelnen Parteien getrennt aufgezählt und verschiedenfarbig dargestellt. Charaktere, die in dieser Kampfrunde eingefügt wurden, bleiben hierbei unberücksichtigt. Zu Beginn der Simulation ist diese Funktion daher

unwirksam.

Simulate x Rounds

Um einen Kampfschauplatz zu simulieren, kann diese Funktion verwendet werden. Hierbei behandelt das *Szenarion* alle Charaktere als NPCs. Die Hot-Keys bleiben aktiv, ein Master-Request wird jedoch nicht mehr angezeigt. Nach Ablauf der eingestellten Simulationsrunden kehrt das Programm zurück zum normalen Simulationsmodus.

6.6 Ende der Simulation

Das *Szenarion* erkennt, wenn nur noch Angehörige einer Partei auf einem Kampfschauplatz am Leben sind. In diesem Fall erscheint ein Requester. Der Meister entscheidet dann über das Abbrechen oder Fortführen der Simulation.

7 Zusammenspiel der Tools

7.1 Allgemein

Die einzelnen Tools der *Szenarios* stehen nicht für sich allein. RFEditor, RKSim und nicht zuletzt der Meister müssen über Schnittstellen miteinander kommunizieren. Das Modulare Konzept des Software-Paketes ähnelt dem eines Compilers. Meister und Software generieren die Anfangsaufstellung, die wie ein Programm vom Simulator gelesen und in eine Handlung übersetzt wird. Im Folgenden werden die einzelnen regelgebenden Komponenten und ihr Platz im Gesamtkonzept vorgestellt:

7.2 Objektgeneration

Die Objekte sind im File `data.plt` gespeichert und werden zu Beginn vom RFEditor und vor jeder Simulation geparkt. Will man die original-Objekte verändern oder neue Objekte erzeugen, so können diese mit einem gängigen Zeichenprogramm gemalt und als Brush abgespeichert werden. Folgendes ist dabei zu beachten:

- *Größe*: 64x64 Pixel.
- *Komprimierung*: KEINE Komprimierung! Die Daten "uncompressed" abspeichern.
- *Speichern*: In das Verzeichnis `gfk/xxx.iff`.
- *Einbinden*: Eigene Objekte, die ein Originalobjekt ersetzen: Struktur im File `config/data.plt` ändern. Den Namen und referenziertes `.iff` ändern. Beim Erzeugen neuer, einen neuen Eintrag im File `config/data.plt` erzeugen. Danach die entsprechenden Flags (`PlantFlagSet`) setzen. Die Syntax ist unbedingt einzuhalten.

Siehe Syntax des Objekt-Beschreibungsfiles

- *Lib.plt*: Sollte der Objektname ebenfalls geändert werden, so muß die Wort-Bibliothek in config/Lib.plt um den entsprechenden Eintrag erweitert werden.
- *Beschränkung*: Gegenwärtig können bis zu 50 Objekte eingelesen werden.

Es ist zu beachten, daß das dargestellte Objekt von vorne in leichter Aufsicht dargestellt werden muß. Die Lichtquelle befindet sich immer hinten links am Rand des Feldes. Achten Sie auch auf die Größenverhältnisse zu den anderen, bereits existierenden Objekten, damit Ihr Zwerg nicht aus versehen auf einen Drachen tritt...

Eine Größe von 30x(-)20 Kästchen sollte Ihr Objekt nicht überschreiten, im Notfall kann dies aber vernachlässigt werden.

Siehe auch PlantFlagSet

7.2.1 Syntax des Objekt-Beschreibungsfiles

Folgende Syntax ist bei der Erzeugung neuer Einträge im File `config/data.plt` einzuhalten:

{Name, {Flag 1, .., Flag n}, Bild}

- Name: String aus Buchstaben, einziges erlaubtes Sonderzeichen: ‘_’. Unter ‘Name’ wird das Bild dann in der Simulation referenziert.
- Flag: Leeres Wort oder Flag aus dem *PlantFlagSet*
- Bild: String aus Buchstaben ohne Sonderzeichen mit Anhang ‘.iff’. Referenziert iff-Bild aus dem gfk/ Verzeichnis. Das File muß dort vorhanden sein, sonst sind weder RKSIm noch RFEditor lauffähig.

ACHTUNG: Alle vor dem Zeitpunkt der Änderung erzeugten Landschaften werden nicht im Bezug auf gesetzte Flags verändert. Alle Veränderungen gelten dann nur für neu zu erzeugende Landschaften. Es ist also unbedingt empfehlenswert,

sich eine Sicherheitskopie des alten Files anzulegen! Werden einfach nur neue Objekte hinten angehängt, so können alte Szenarios problemlos weiterbenutzt werden (Höchstgrenze 50 Objekte!).

7.3 Das PlantFlagSet

Das **PlantFlagSet** ist die Menge aller Flags, die ein Feld eines Kampfschauplatzes annehmen kann. Durch setzen von Flags während der Konstruktion durch den RFEEditor oder in der Simulation durch das Meister-Menü entscheidet der Meister, welche Felder passierbar sind, welche Türen verschlossen und auf welchen Feldern die Spieler Text-Hinweise oder Beschreibungen bekommen.

Normalerweise ist kein Flag gesetzt. Erst, wenn ein Objekt auf ein Feld aufgebracht wird, werden die Flags, die zu dem Objekt¹⁴gehören, auf das entsprechende Feld übertragen. Nachträglich können diese wieder verändert werden (*siehe Modi*). Es stehen die folgenden Flags zur Verfügung:

BlockWay, BlockNormalWay, BlockView, BlockMissles, BlockSomeMissles, SlowDown, Hidden, LeadsTo, Open, Close, Trap, Story, Character.

7.3.1 Die Flags

BlockWay

Das Feld wird gesperrt, eine Spielfigur kann es nicht passieren. Durch Setzen von Mauerfeldern mit dem RFEEditor wird dieses Flag automatisch gesetzt. Auch alle Objekte sollten dieses Flag beinhalten.

BlockView

Ein Charakter kann nicht hinter ein Feld mit dem gesetzten Flag BlockView sehen. Dies impliziert natürlich auch, daß er keine Informationen über Felder bekommt, die durch dieses Feld verdeckt werden.

BlockMissels

Alle Geschosse prallen an solch einem Feld ab. Bisher ist dieses Flag nicht in der Simulation implementiert.

¹⁴Ausnahmen sind Mauern, die, sobald das Feld abgespeichert wird, mit Sperr-Flags versehen werden.

BlockSomeMissels

Es ist besonders schwer, durch dieses Feld zu schießen (z.B. Fenster, Baum...). Bisher ist dieses Flag nicht in der Simulation implementiert.

SlowDown

Es benötigt mehr Zeit dieses Feld zu überqueren, z.B. wenn ein Zauberspruch auf diesem Feld aktiv ist. Steigungen verzögern automatisch die Bewegung in der Simulation, daher muß dieses Flag nur dann von Hand gesetzt werden, wenn eine zusätzliche Verlangsamung erwünscht ist (z.B. Sumpf).

Bind

Solche Felder bilden Verbindungen zu anderen. Alle Felder, die dieses Flag enthalten, aber nicht durch **BIND** erzeugt wurden, können zum Verlassen des Spielfeldes benutzt werden. Bisher wird dies vom RKSimulator nicht unterstützt.

Open/Close

Türen kann man öffnen oder schließen, aber auch z.B. den Eingang zu einem Baumhaus oder einem magischen Portal. Um ein verschlossenes Tor oder eine Tür zu öffnen, benötigt ein Charakter einen hohen Wert in "*öffnen*". Nicht nur Spieler-Charaktere, sondern alle anwesenden Kreaturen können Türen öffnen (oder dies zumindest versuchen). Offene Türen können ohne Schwierigkeiten passiert werden.

Trap

Fallen werden automatisch im Allfile eingetragen und müssen später dort noch genauer definiert werden, d.h., daß dort eine Falle aus dem Verzeichnis **Traps/** eingesetzt werden muß. Bisher wird dies vom RKSimulator nicht unterstützt.

Story

Dieses Flag dient zur Beschreibung der erzeugten Landschaft. Besondere landschaftliche Gegebenheiten können so während der Simulation geschildert werden, wenn ein Charakter sich in der Nähe oder auf einem Feld befindet, das dieses Flag trägt. Das Allfile enthält all diese Felder, die dann dort vom Meister genauer

definiert werden müssen, falls mehr als eine Standardmeldung im RKSimulator auftauchen soll. *Siehe auch Allfile.*

Character

Der Eintrag bedeutet, daß hier ein Monster/Charakter zu Beginn der Kampfsimulation steht. Der Eintrag wird an das Allfile weitergereicht. Dort muß ein existierender Charakter aus dem Verzeichnis **Character/** eingesetzt und modifiziert werden. Geschieht dies nicht, so wird dieses Flag zu Beginn der Simulation gelöscht. *Siehe auch Charakter-File.*

7.3.2 Beispiel

Nehmen wir an, der Meister wollte sein Objekt “Troll” in das *data.plt*-File einbinden und grübelt nun, welche Flags des PlantFlagSets zu setzen sind...

Nun, er findet, niemand solle einfach über seinen mit viel Mühe und Zeit gemalten Troll hinweggehen. Er setzt somit das Flag *BlockWay*. Eine weise Wahl. Steht der Troll im Spiel in einem Gang (2m breit), so kommt jeder Spieler/ Nicht-Spieler in den Genuß der extremen Körperkräfte dieses Untieres...

“Trolle sind ja nicht gerade winzig!”, spricht unser Meister, also sollte man nicht auf Felder sehen können, die sie aufgrund ihrer enormen Ausmaße verdecken. Unser Meister setzt das Flag *BlockView*. Als Folge sehen seine Charaktere nicht, daß im Gang, direkt hinter dem Troll, noch fünf weitere warten¹⁵.

Genug Flags, denkt sich der Meister, und startet die Simulation... Entsetzt stellt er fest, seine Mega-Monster stehen leblos im Szenario herum. Er sieht im *characters/* Verzeichnis nach, die Charakter-Daten liegen aber vor. Was hat er falsch gemacht? Natürlich hat er vergessen, daß Flag *Character* zu setzen. Das ist schnell erledigt. Wenige Minuten später stehen seine Trolle bereit, jedem Eindringling, der ihren Stollen betritt, bei der Lösung des Rätsels (siehe Fußnote) zu helfen...

Seine Überlegungen führen zu folgendem Ergebnis:

(Troll,{BlockWay,BlockView,Character},Troll.iff)

¹⁵Hätten sie besser das Rätsel auf dem verwitterten Schild am Eingang gelöst: ‘Vorsicht: Xollen Xoller Xolle’ = Vorsicht: Stollen voller Trolle...

7.4 Das Allfile

7.4.1 Allgemein

Das Allfile¹⁶ wird erzeugt, sobald ein Feld mit dem RFEEditor abgespeichert wird. Es ist mehr als ein einfacher Speicher zum Datenaustausch der einzelnen Tools. Kurz, es ist sozusagen die **Dokumentation** der Landschaft, auf dem Rollenspiel stattfinden soll. Das Allfile bietet dem Meister die Möglichkeit, das Szenario mit Leben zu füllen, eigene Ideen einzubringen und so eine Atmosphäre zu schaffen, die gutem Rollenspiel gebührt — weder Spieler noch Monster verdienen reines “Hack & Slash”...

Das Allfile wird automatisch generiert, und “leblos” in das allfile/ Verzeichnis geschrieben, mit dem gleichen Namen, den auch die erzeugte Karte trägt. Lediglich der Anhang .alf kennzeichnet es als Dokumentation. In diesem Zustand kann das Simulations-Tool des *Szenarions* noch keinen Kampf durchführen. Elementare Daten (welche Charaktere sind aus dem *characters/* Verzeichnis einzubinden) und Meistergenerierte Informationen (Welche Besonderheiten sind den Charakteren zu schildern) **müssen** noch eingegeben werden. Das Allfile ist in vier Bereiche gegliedert:

- **Kopf:** Der Kopf beschreibt die Karte und enthält Namen und Feldgröße. Hier sind keine Einträge vorzunehmen, der Meister kann hier aber eigene Notizen hineinschreiben.
- **Story:** Diese Sektion wird durch das Schlüsselwort *Story:* eingeleitet. Hier können besondere landschaftliche Gegebenheiten, Orte, Gegenstände oder was immer dem Meister als wichtig erscheint, geschildert werden.
- **Traps:** Beginnend mit dem Schlüsselwort *Traps:* endet die Story-Sektion und beginnt die Beschreibung der sich auf dem Kampfschauplatz befindenen Fallen. Bisher ist eine Nutzung von Fallen nicht implementiert.

¹⁶Original-Idee von M.Zimmermann

- **Characters:** Diese Sektion legt fest, welche der Charaktere aus dem characters/ Verzeichnis an der Simulation teilnehmen. Zusätzlich bietet es dem Meister noch die Möglichkeit, die Charaktere kurz zu beschreiben.
ACHTUNG: Die Reihenfolge der Charaktere einer Partei ist nicht ebenfalls wichtig! Potentielle Anführer sollten vor Gefolgsleuten angeordnet werden, da Gefolgsleute (Verhaltensflag “obeye”) ohne Führungspersönlichkeit schlecht oder gar nicht mehr kämpfen.

7.4.2 Syntax

Grundsätzlich kann der Meister in allen Sektionen weitere Einträge einfügen, die durch den RFEEditor noch nicht erzeugt wurden. Hierzu hat er sich jedoch strikt an die von der Sektion abhängigen Syntax zu halten. Kommentare sind nicht erlaubt. Die Beschreibung ist aus der Sicht der Charaktere vorzunehmen (siehe Beispiel). Die Charaktere selbst erhalten während der Simulation die Information über ein Feld, die ihrem Wahrnehmungswurf entsprechen:

- **Story:** In jedem Allfile gibt es einen, das allgemeine Szenario beschreibenden Eintrag (*About*). Danach folgen die von der Landschaft abhängigen Einträge unter den jeweiligen Namen der Objekte oder Felder (*empty*), die das Plant-Flag *Story* tragen.

```
( Name,X-Koordinate,Y-Koordinat17,
  always { allgemeine Beschreibung des Felds. },{ },
  fail { Zusatz, falls der Wahrnehmungswurf scheiterte. },{ },
  normal { Zusatz, falls der Wahrnehmungswurf erfolgreich war. },{ },
  good { Zusatz bei sehr gut gelungenem Wahrnehmungswurf. },{ }
)
```

- **Characters:** Hier wurde für jedes Vorkommen des PlantFlags “*Character* ” ein Eintrag erzeugt. Nicht erwünschte Einträge können gelöscht, eigene später hinzugefügt werden.

¹⁷Koordinaten des Feldes/ Objektes auf der Karte, Ursprung ist die untere linke Ecke.

Hier wird auch festgelegt, welche Charaktere Spieler-Charaktere (**PlayerCharacter**) oder einfache computergesteuerte Charaktere (**NonPlayerCharacter**) sind. Eine Sonderrolle nehmen hierbei **DisplayNPCharacter** ein. Dies sind Charaktere, die den Spieler-Charakteren der eigenen Partei als Ikone angezeigt werden (z.B. fünfter Spieler-Charakter).

```
( Charakter-Name, X-Koordinate, Y-Koordinat,
  Charakter-Art18, { Verhaltensflags19 },
  always { allgemeine Beschreibung des Charakter. }, {},
  fail { Zusatz, falls der Wahrnehmungswurf scheiterte. }, {},
  normal { Zusatz, falls der Wahrnehmungswurf erfolgreich war. }, {},
  good { Zusatz bei sehr gut gelungenem Wahrnehmungswurf. }, {}
)
```

Siehe auch Verhaltensflags

ACHTUNG: Zum jeweiligen Namen muß ein Eintrag im File config/Lib.plt vorhanden sein oder erzeugt werden, bevor die Simulation gestartet wird.

¹⁸PlayerCharacter, NonPlayerCharacter oder DisplayNPCharacter

¹⁹neutral, attackOnSight, stand, join, inform oder obeye

7.4.3 Allfile-Beispiel

Unser imaginärer Meister schreitet wieder zur Tat: Er hat mit dem RFEEditor seine erste Landschaft erzeugt. Nun will er sein Allfile Lichtung.alf im Verzeichnis allfile/ ausfüllen:

Seine Landschaft besteht aus einer großen Lichtung, umgeben von Wald. In der Mitte befindet sich ein steinernes Portal, das durch schwarze Magie verzaubert wurde. Jeder, der es betritt, wird in den Kerker des Erz-Bösewichtes teleportiert (der Meister entfernt den entsprechenden Charakter durch das *Meister-Menu*).

Um Vorbeiziehende zu warnen, hat der Meister den Nicht-Spieler-Charakter Ruhl, einen alten Bauern, an das Portal gestellt.

Greifen die Spieler nun einfach den armen alten Ruhl an, so werden sie nie erfahren, daß sie das Portal besser nicht betreten sollten... Nicht immer muß und sollte das *Szenarion* also zum blutigen Gemetzel verwendet werden!

Der Meister überlegt, wie er das Szenario beschreiben soll. In der *About* Information schildert er kurz die allgemeine Lage und gibt den Charakteren einige Tips, je nach ausfallendem Wahrnehmungswurf...

Danach füllt er den *Story*-Block für das Portal aus. Die Charaktere, die die Ruhnen des Portals falsch deuten, sollen annehmen, das Tor führe direkt in eine Schatzkammer²⁰.

Nun werden die Charaktere definiert. Ruhl soll einfach nur am Portal stehen und sich nicht bewegen, jeden Fragenden aber warnen. Jeder verwirrte Charakter soll in Ruhl einen tückischen Feind sehen...

Startet nun der Meister die Simulation durch das Tool RKSIm und sind alle referenzierten Charakter-Files und Standard-Bibliotheken vorhanden und fehlerfrei, so

²⁰Unser Meister freut sich schon auf den Streit, der zwischen den Spielern entsteht, falls sie die Ruhnen unterschiedlich interpretieren.

wird seine Lichtung wenige Sekunden später zu Leben erwachen... Das fertig ausgefüllte Allfile (die Spieler-Charaktere wurden aus Platzgründen nicht aufgeführt) aus dem Beispiel könnte dann so aussehen:

Allfile-Beispiel:

```
Map      : Lichtung
Size     : 50 x 50

Story    :

(About,0,0,
always {Ihr seid auf einer kleinen Lichtung, in deren Mitte ein steinernes Portal steht,
davor kann man eine Gestalt erkennen.},{},
fail    {Gefahr geht nur von der Gestalt aus, das Portal ist ungefährlich},{},
normal  {Das Portal scheint magisch zu sein, die Gestalt ist ein alter Mann},{},
good    {Das Portal kommt Dir merkwürdig vor, irgendwas ist daran nicht geheuer. Ein
alter Mann blickt Dich sorgenvoll an.},{})

(Portal, 21, 27,
always {Ein riesiges steinernes Portal, überseht mit mystischen Runen.},{},
fail    {Aus einer Rune erkennst Du, es führt zu einem großen Schatz!},{},
normal  {Die Runen sagen Dir, daß es ratsam wäre, es nicht zu betreten},{},
good    {Aus den Runen erkennst Du, es ist eine magische Falle},{})

Traps    :

Characters :

(Ruhl, 20, 27, NonPlayerCharacter, {stand,neutral,inform},
always {Ein alter Mann in merkwürdigen Gewändern.},{},
fail    {Er muß ein Schwarz-Magier sein, der Dich töten will!},{},
normal  {Es scheint, als wolle er mit Euch reden.},{},
good    {Es scheint, als wolle er Euch warnen. Es muß etwas mit dem Protal zu tun haben!},{})

(..)
```

7.5 Verhaltensflags und NPC-Strategien

7.5.1 Die Flags

Die Verhaltensflags bestimmen das Handeln der NPC's (Nicht-Spieler-Charaktere) in der Simulation. Die Flags können einzeln oder zusammen gesetzt werden und gelten für jeden Charakter nur zu Beginn der Simulation. D.h., sie können sich situationsbedingt nach jeder Runde ändern (Ein Charakter, der sich ergibt, setzt nur die Flags {neutral, join}).

Zwei Flags besitzen eine höhere Priorität als die übrigen: **neutral** und **stand**. So würde folgender Charakter nur angreifen, falls er selbst angegriffen wird: {attackOnSight, neutral } oder {attackOnSight, stand }. Die Bedeutung der Flags im einzelnen:

- **neutral:** Der Charakter verhält sich so lange passiv, bis er angegriffen wird. Danach kämpft er (siehe attackOnSight)!
- **attackOnSight:** Sobald der NPC einen Charakter wahrnimmt, der nicht zu seiner Partei gehört, greift er ihn an.
- **stand:** Der Charakter bleibt an einem Ort stehen, so lange er nicht direkt angegriffen wird. (z.B. eine Wache, Gegensatz zu attackOnSight)
- **join:** Der Charakter läßt sich leicht von einem anderen Charakter überzeugen, mit ihm zu kämpfen.
- **inform:** Der Charakter gibt leicht seine in den Charakter-Daten festgelegten Informationen preis.
- **obey:** Der Charakter gehorcht dem im Allfile vor ihm stehenden Charakter, der zur gleichen Partei wie er gehört und durch kein "obey"-Flag gekennzeichnet ist. So lange es zu keinem Kampf kommt, folgt er seinem Anführer auf Schritt und Tritt.

Findet er keinen Anführer (z.B. durch dessen Tod), so ist er eine Runde lang handlungsunfähig und kämpft danach selbstständig (`attackOnSight`) oder ergibt sich (`join,neutral`), je nach seinen Charakter-Daten. Es empfiehlt sich also, die Charaktere in einer gewissen Rangfolge im Allfile aufzuführen, also die abhängigen Charaktere hinter den potentiellen Anführern.

7.5.2 NPC-Strategie

Jeder nicht bewußtlose lebende Nicht-Spieler-Charakter (NPC) durchläuft eine Entscheidungskette. Zuerst wird festgestellt, ob er überhaupt (noch) kampffähig ist oder eine gewisse Wahrscheinlichkeit besteht (je nach *Charakter-Wert Ergeben*), sich zu ergeben.

Grundsätzlich stehen auch ihm die Standard-Handlungen zur Verfügung. Will der NPC angreifen (je nach gesetzten Flags), so betrachtet er alle für ihn sichtbaren Gegner: Befinden sich bereits Gegner mit ihm im Nahkampf, so wählt auch er den Nahkampf, je nach Situation stehen ihm Rundumschlag, Abdrängen und Angriff zur Verfügung.

Andernfalls wählt der NPC mit hoher Wahrscheinlichkeit eine Angriffsart, die ihn nicht in Nahkämpfe verwickelt, sofern er zu Fernkampf oder Einsatz von Magie fähig ist.

Dabei verteidigen sich angegriffene Charaktere best möglich, sofern sie ihren Gegner wahrnehmen.

7.5.3 Beispiele

Suchen wir wieder unseren imaginären Meister während des Ausfüllens des *Allfiles* auf. Er möchte in sein Szenario neben Ruhl (*siehe Beispiel Allfile*) noch drei weitere Charaktere hinzufügen, die auf der anderen Seite der Lichtung am Ausgang den Charakteren auflauern: Die Schwarzelben Mortash und die Brüder Imrith und Imerith. Mortash und Imerith beherrschen ihren Bogen perfekt (+90/+79), Imrith, der ältere und erfahrenste NPC lauert hinter einer Biegung des Weges mit seinem Langschwert. Von der Lichtung aus kann man weder sie sehen, noch können sie auf

die Lichtung blicken. Imerith und Mortash stehen ca. sechs Meter entfernt hinter Imrith auf dem vier Meter breiten Weg und können so die Felder vor Imrith bis zur Biegung gut unter Feuer nehmen.

Wie kann der Meister nun erreichen, daß in der Simulation die Schwarzelben angreifen, sobald ein gegnerischer Charakter um die Ecke biegt?

Die Lösung ist recht einfach: Imrith bekommt das Flag `{attackOnSight}`, d.h. er stürmt auf seinen Gegner zu, sobald er um die Ecke biegt. Die beiden anderen werden mit dem Flag `{obeye}` ausgerüstet, so nehmen sie den Gegner unter Feuer, der durch ihren Anführer Imrith attackiert wird. Solch ein Hinterhalt endet mit hoher Wahrscheinlichkeit tödlich für den zuerst angegriffenen Spieler-Charakter, den der Hinterhalt unvorbereitet trifft.

Der Meister hat in der Simulation festgestellt, daß dieses Setting zu schwer für seine unerfahrenen Charaktere war (keiner seiner drei Helden überlebte). Er will jetzt, daß Imrith die Charaktere auffordert, ihm Geld zu geben²¹, dazu setzt er Imrith so, daß er den Weg versperrt (Flags: `{inform,neutral}`). Imertith und Mortash dürfen so durch ihr gesetztes `{obeye}`-Flag erst feuern, sobald Imrith angegriffen und so sein `{neutral}`-Flag gelöscht wird (oder der Meister dieses Flag vom *Meister-Menu* aus löscht).

²¹Wird er angesprochen, so sagt er: "Gebt mir 10 Goldstücke, so könnt Ihr passieren!"

7.6 Charakter-File

Die Charaktere sind durch Daten-Files beschrieben, die sich im characters/ Verzeichnis befinden. Das File ähnelt stark einem in jedem Rollenspiel üblichen Charakter-Bogen. Ich denke, daß einem Meister einer Rollenspielgruppe Schlagworte wie *Charakter-Werte* und *Würfe/Proben* bekannt sind...

7.6.1 Eigenschaften und Würfe

Prozent	Eigenschaften	W100	W20
1%..5%		1..5	1
6%..10%	-10	6..10	2
11%..15%		11..15	3
16%..20%	-5	16..20	4
21%..25%		21..25	5
26%..30%		26..30	6
31%..35%		31..35	7
36%..40%		36..40	8
41%..45%		41..45	9
46%..50%	0	46..50	10
51%..55%		51..55	11
56%..60%		56..60	12
61%..65%		61..65	13
66%..70%		66..70	14
71%..75%		71..75	15
76%..80%		76..80	16
81%..85%	+5	81..85	17
86%..90%		86..90	18
91%..94%	+10	91..94	19
95%..97%	+15	95..97	20
98%..99%	+20	98..99	20
100%	+25	100	20
101%	+30	101	21
102..%	+35	102+	21+

Im *Szenarion* wird natürlich nicht gewürfelt, sondern ein (speziell initialisierter) Zufallsgenerator verwendet, der Ergebnisse analog zum Wurf mit einem 100-seitigen Würfel liefert²². Zu dem Ergebnis des Wurfes wird der jeweilige Wert der Eigenschaft addiert, die getestet werden soll (auch Probe genannt). Da nicht alle auf dem Markt befindlichen Rollenspiele ihre Charakter-Eigenschaften in Prozenten (normalerweise 0% bis 100%) angeben, ist es nötig, diese Normierung beim Ausfüllen des Charakter-Files vorzunehmen. Das folgende Beispiel möge dies veranschaulichen:

Unser Meister spiele das Rollenspiel eines Herstellers, dessen Cover saubere muskelbepackte Superglanzhelden Marke *Muskulus* zieren. Angriffswürfe, Paraden, Proben u.s.w. werden mit einem W20 ermittelt. Wie kann nun Held Muskulus in das

²²Es gibt natürlich auch offene Würfe, deren Ergebnisse rein theoretisch zwischen $-\infty$ und $+\infty$ liegen können

Szenarion aufgenommen, bzw. seine Charakter-Eigenschaften normiert werden? Nun, dies ist recht einfach: Man interpretiert das Maximalergebnis eines Wurfes mit W20 als 100%, eine eins als 5%. Dazu geht man nach der anfangs aufgeführten Tabelle vor.

ACHTUNG: Alle Talente oder Eigenschaften mit Wert Null (oder weniger) sollten im *Szenarion* als **-25** behandelt werden.

Dabei beziehen sich die Prozentangaben auf die Werte für Talente, also Fähigkeitswerte, die z.B. das Öffnen von Schlössern beschreiben. Eigenschaften, also Grundwerte wie *Stärke*, *Intelligenz* und *Geschicklichkeit*, werden im *Szenarion* nicht prozentual angegeben, sondern durch einen Bonus (siehe Spalte für Eigenschaften).

Die Tabelle kann nach oben und unten ergänzt werden und dient nur als Anhaltspunkt. Letztendlich bleibt es jedem Meister selber überlassen, seine Charaktere einzustufen.

Allgemein kann man sagen, Talente geben die prozentuale Wahrscheinlichkeit an, mit der eine Aktion gelingt, wobei auch Werte über 100% und geringer als 0% zulässig sind.

Beispiel

Hat Muskulus einen Intelligenz-Wert von 8, so kann der Meister diesen Wert nach eigenem Ermessen in 36..40 wandeln. Sein Bonus für jeden Intelligenz-Wurf ist nun +0.

In seinem langen Dasein als Held entwickelte Muskulus jedoch ein gewisses Geschick für das Öffnen von Schlössern. Sein Wert in diesem Talent liegt bei 16 (von 20). Im *Szenarion* entspricht dies einer 76..80.

Die Devise dieses Helden ist: "Ruckzuck ist die Fresse dick!". Dementsprechend hoch ist sein Attacke-Wert: 17. Auch dies wird wieder prozentual umgerechnet auf 81..85.

7.6.2 Syntax

Die jeweilige Syntax wird noch genauer anhand der einzelnen Charakter-Werte definiert. Es sei darauf aufmerksam gemacht, daß im Charakter-File weder Kommentare, noch Sonderzeichen erlaubt sind und die den jeweiligen Eigenschaften und Talenten entsprechenden Regeln genauestens eingehalten werden müssen.

Grundsätzlich gibt es fünf verschiedene Syntax-Typen:

Typ 1:

Schlüsselwort, *String*

Typ 2:

Schlüsselwort, *Wert*

Typ 3:

Schlüsselwort, *String*, *Wert*

Typ 4:

Schlüsselwort, *String*₁, *String*₂, (*..*), *String*_{n-1}, *String*_n

Typ 5:

Schlüsselwort, *String*₁, *Wert*₁, *String*₂, *Wert*₂, (*..*), *String*_n, *Wert*_n

Ein **String** ist eine ASCII-Folge von Buchstaben {a..z, A..Z, äöüß, ÄÖÜ} und Zahlen {0..9}, sowie den folgenden Sonderzeichen: {'-', '?', '!', ':', ';', '}'}. Statt einem Space muß das Sonderzeichen '_' verwendet werden, das in Simulationstexten als Space dargestellt wird. Bei **Werten** sind numerische Werte {-32000..32000} zulässig. Das ASCII-Zeichen Space ist generell unzulässig!

„Uns finden sie auch im *characters*/-Verzeichnis! Wir sind die Helden des Zyklus: (v.l.n.r.) Gronar, Gurth, Hawk, Huma, Morgai, Chiage (unsichtbar!)“

7.6.3 Die Daten

Der Meister muß die Datenfiles für seine Charaktere mit großer Sorgfalt ausfüllen, denn ein Gnom, der es an Körperkräften mit einem Drachen aufnehmen kann, wird während der Simulation gewiß für Aufsehen sorgen! Falls sie weitere Beispiele suchen, sehen sie einfach im *characters*/-Verzeichnis nach. Dort finden sie auch die oben abgebildeten fünf Charaktere.

Das folgende Beispiel-Charakter-File soll als Vorlage dienen. In ihm sind alle vom *Szenarion* berücksichtigten Eigenschaften aufgeführt:

Ein typisches Charakter-File:

***** Gronar.char *****

Name,Gronar
Beruf,Anemist
Rasse,Zwerg
Anatomie,Mann
Stufe,3
Partei,Gruppe
Bild,Zwerg
Stärke,+5
Intelligenz,0
Geschicklichkeit,+20
Geschwindigkeit,3
Geschosse,10
Schild,+55
Helm,0
Rüstung,ohne_Rüstung
B&M,+35
Nahkampfwaffe,Streitaxt,+55
Fernkampfwaffe,Bola,+25
Schwimmen,+55
Verbergen,+10
Öffnen,+5
Sprüche,+45
Wahrnehmung,+17
Körperentwicklung,32
Lebenspunkte,21
Magiepunkte,9
Auftreten,+0
Defensivbonus,+80
WWMagie,+40
Ergeben,5
Spruchlisten,Feueraxt,TiefschlafI,FremdheilenI,SelbstheilenI,VerhüllenI,SchutzI
Waffen,Kriegshammer,+15,Faust,+25
Gegenstände,Seil,Fackel
Wissen,Ich_habe_die_Hallen_Khazad_Dhums_gesehen.
Kampfschrei,Khazad_Duhm!

Die Daten eines Charakters lassen sich in folgende Oberbegriffe aufgliedern. Zum Vergleich kann der Beispiel-Charakter Gronar herangezogen werden:

Eckdaten:

Eckdaten kennzeichnen den beschriebenen Charakter und drücken nicht unmittelbar seine Fähigkeiten auf einzelnen Gebieten aus:

- **Name:** Unter dem hier angegebenen Namen wird der Charakter vom *Szenarion* verwaltet. Feindliche Charaktere bekommen jedoch nur die Rasse eines Gegners mitgeteilt.
Typ 1 - Syntax.
- **Beruf:** Erzählt ein Charakter freizügig von sich, so teilt er auch seinen "Beruf" mit. Man kann es auch als Charakter-Klasse interpretieren.
Typ 1 - Syntax.
- **Rasse:** Kaum ein Mensch kann auf Anhieb den Namen eines Passanten erraten. Genau so geht es den (N)PCs im *Szenarion*, die ihre Gegner als Zugehörige einer Rasse wahrnehmen.
Typ 1 - Syntax.
- **Anatomie:** Die Anatomie eines Charakters dient zur Darstellung der getragenen Rüstungsteile und Waffen während der Simulation. Mögliche Einträge sind {Mann, Frau}²³
Typ 1 - Syntax.
- **Stufe:** Rettungswürfe und Einschätzung eines Gegners werden durch die Stufe eines Charakters modifiziert. Da verschiedene Rollenspiele den Aufstieg eines Charakters unterschiedlich behandeln, ist die Stufe/ der Level nur ein Richtwert.
Typ 2 - Syntax.

²³Natürlich führt das bei Monstern zu Konflikten. Ich bitte dies zu verzeihen...

- **Partei:** Dieser Eintrag ist von elementarer Bedeutung: Er legt die Gruppenzugehörigkeit fest. Charaktere gleicher Partei sind Freunde, Charaktere unterschiedlicher Parteien betrachten sich als (potentielle) Gegner. Durch die Aktion *Ergeben* kündigt ein NPC seine Bereitschaft an, die Partei zu wechseln.
Typ 1 - Syntax.
- **Bild:** Der hier angegebene String bestimmt, welche Grafik aus dem gfk/ Verzeichnis zur Darstellung des Charakters verwendet wird. Der Anhang ".iff" darf nicht benutzt werden. Das Objekt muß auch im File *config/data.plt* unter dem gleichen Namen vorhanden sein!
Typ 1 - Syntax.

Eigenschaften:

Bei den Eigenschaften eines Charakters handelt es sich um die Grunddaten, die z.B. einen Magier von einem Krieger unterscheiden (Intelligenz/Stärke). Diese Daten modifizieren in vielen Rollenspiel-Systemen auch die Talent-Werte.

- **Stärke:** Sie ist ein Maß für körperliche Leistungsfähigkeit. Im *Szenarion* bestimmt sie auch die Fähigkeit, einen Gegner abzudrängen.
Typ 2 - Syntax.
- **Intelligenz:** Eine hohe Intelligenz erleichtert den Umgang mit Zaubern und legt oft auch die Zahl der Magiepunkte fest.
Typ 2 - Syntax.
- **Geschicklichkeit:** Bestimmt den Umgang mit Waffen und erleichtert die Nutzung vieler Talente.
Typ 2 - Syntax.
- **Geschwindigkeit:** Hierbei handelt es sich um die Grundreichweite eines Charakters in Feldern pro Runde. Folgende Überlegung dient als Bemessungsgrundlage: Menschen bewegen sich mit ca. 3, Pferde mit 6, Vögel mit 8 (..) Feldern pro Runde. Weitere Monster sind, entsprechend den eigenen Vorstellungen, einzupassen.

In die Reichweite fließt auch der zu der jeweilig getragenen Rüstung gehörende Bewegungs-Bonus ein.

Typ 2 - Syntax.

Rüstung und Modifikation:

Nun zu den getragenen Rüstungen bzw. Rüstungsteilen. Zusätzlich zu der eigentlichen Rüstung, die den Torso eines Charakters schützt, kann sich der Charakter mit zusätzlichen Teilen ausrüsten (z.B. Helm), die ggf. den Defensivbonus erhöhen²⁴ und/oder Auswirkungen bestimmter kritischer Treffer mildern können. Im Charakter-Bogen dürfen nur die Rüstungsteile aufgeführt werden, die der Charakter auch im Augenblick des Kampfes trägt (so trägt Gronar nur einen Helm (DB +0) und einen Schild (DB +55), aber z.B. keine Beinschienen). Wird ein bestimmter Gegenstand zerstört, so wird der Defensivbonus um den jeweiligen Wert vermindert. Monster, die keine Rüstungen tragen können, werden aufgrund der Beschaffenheit ihrer Haut eingeordnet. So ähnelt die Haut eines Drachens einem Plattenpanzer.

- **Rüstung:** Dieses Schlüsselwort muß im Charakter-File vorhanden sein. Es legt die in der Simulation getragene Torso-Rüstung fest. Rüstungen behindern einen Charakter in seiner Bewegungsfreiheit. Der Bewegungs-Bonus für die getragene Rüstung sollte daher durch den Bonus für die Eigenschaft Stärke²⁵ oder Geschicklichkeit²⁶ (7.6.1) plus mögliche Erfahrungswerte (je nach Stufe und Beruf) festgelegt werden. Zwischen folgenden Rüstungen kann gewählt werden: { Plattenpanzer, Kettenpanzer, verstärkte_Lederrüstung, Lederrüstung, ohne_Rüstung }

²⁴je nach Güte kann der Meister den Defensivbonus um einen Wert erhöhen.

²⁵für Ketten- und Plattenpanzer

²⁶normale Kleidung, Leder- und verstärkte Lederrüstung

“*Plattenpanzer*”: Er ist die schwerste Rüstung und bietet den besten Schutz vor schwersten Treffern, macht den Träger aber recht unbeweglich. Der Bewegungs-Bonus wird durch die Eigenschaft Stärke (s.o.) bestimmt und um -60 erniedrigt.

“*Kettenpanzer*”: Diese Rüstung ist leichter als ein Plattenpanzer, sie bietet Schutz gegen schwere Schläge und erlaubt eine akzeptable Bewegungsfähigkeit. Der Bewegungs-Bonus wird durch die Eigenschaft Stärke (s.o.) bestimmt und um -45 erniedrigt.

“*verstärkter Lederpanzer*”: Solch ein Panzer besteht aus gehärtetem Leder mit aufgesetzten Metall-Nieten an kritischen Bereichen. Er bietet Schutz gegen mittlere Schläge und erlaubt eine kaum eingeschränkte Bewegungsfähigkeit. Der Bewegungs-Bonus wird durch die Eigenschaft Geschicklichkeit (s.o.) bestimmt und um -30 erniedrigt.

“*Lederrüstung*”: Dieser Schutz ähnelt der verstärkten Lederrüstung, ist jedoch ohne Metall-Verstärkungen und dadurch leichter, bietet jedoch nicht den gleichen Schutz. Der Bewegungs-Bonus wird durch die Eigenschaft Geschicklichkeit (s.o.) bestimmt und um -15 erniedrigt.

“*ohne_Rüstung*”: Normale Kleidung fällt unter diesen Oberbegriff. Die Bewegung wird nicht eingeschränkt. Der Bewegungs-Bonus wird durch die Eigenschaft Geschicklichkeit (s.o.) bestimmt.

Typ 1 - Syntax

- **B&M**: Dieser Bewegungs-Bonus legt die Wendigkeit eines Charakters fest. Besitzt ein Charakter einen hohen Wert, so erhöht sich auch seine Reichweite/Runde und die Reaktionszeit (im *Szenarion* gleichbedeutend mit der Reihenfolge der Angriffe). Dieser Wert muß auch durch die aktuell getragene Rüstung (s.o.) und eventuelle zusätzliche Rüstungsteile vom Meister modifiziert werden.

Beispiel: Würde Gronar nicht rüstungslos kämpfen, sondern mit einem Plattenpanzer, so wäre sein B&M-Wert nicht +35, sondern -25, da ihn die schweren Metallplatten enorm in seiner Bewegung behindern. Als Folge würde sich seine

Reichweite (Felder, die er pro Runde zurücklegen kann) und seine Reaktionszeit verschlechtern...

Typ 2 - Syntax

- **Rüstungsteile:** Die folgenden Schlüsselworte für zusätzliche Rüstungsteile können optional verwendet werden. D.h., trägt ein Charakter sie im Kampf, so müssen sie im Charakter-File eingetragen werden, sonst dürfen sie dort nicht stehen. Der Defensivbonus kann durch gut verarbeitete oder magische Rüstungsteile erhöht werden (Modifikation (Wert) hinter dem Schlüsselwort). Folgende Zusatzausrüstung wird unterstützt: { Armschiene, Beinschiene, Helm, Plattenschurz, Schild, Schwebescheibe }

“Armschiene” & “Beinschiene”: Sie dienen dem Schutz der Arme und Beine vor Schlägen, die mit dem Schild nicht pariert werden konnten, und können von unterschiedlicher Qualität sein. Einfache Schienen mildern nur bestimmte kritische Treffer. Schienen behindern in der Regel die Bewegung (je -5 auf B&M).

“Helm”: Ob nun aus Leder oder Metall, er verhindert einige kritische Kopftreffer, behindert aber die Sicht (-5 auf Wahrnehmung). Auch hier hängt eine eventuelle Erhöhung des Defensivbonus von der Qualität des Objektes ab.

“Plattenschurz”: Er schützt den Becken-, Hüft- und Genitalbereich vor unangenehmen Auswirkungen schwerer Treffer auf entsprechende Körperteile.

“Schild”: Den Schild gibt es in den verschiedensten Kulturen. Er kann aus Leder, Holz oder Metall bestehen, kann in der Hand getragen, am Oberarm befestigt oder umgeschnallt im Kampf verwendet werden. Somit ist auch der Einsatz von Zweihänderwaffen gleichzeitig gestattet. Der Offensivwert sollte aber entsprechend vermindert werden. Ein normaler Schild erhöht den Defensivbonus um +25. Gronars Schild stammt aus den Hallen Morias, seine

meisterliche Verarbeitung verleiht ihm einen Schutzfaktor von +55!

“*Schwebescheiben*”: Sie sind durch Lederriemen an der Schulterpanzerung befestigt. Sie sollen die Nahtstellen zwischen Armschienen und Brustharnisch schützen. Einige kritische Treffer auf den Schulterbereich werden gemildert.

Typ 2 - Syntax

Waffen und Angriffswerkzeuge

Dieser Teil widmet sich der Benutzung einzelner Waffen. Eine Grundidee ist hierbei, daß es verschiedene Klassen von Waffen gibt. Ein Charakter beherrscht alle Waffen einer Klasse mit dem gleichen Geschick und kann seine Fähigkeit pro Stufe für die Klassen getrennt erhöhen. Daher muß für jede Waffe einzeln der Fähigkeitswert mit angegeben werden. Dabei können auch Rollenspiele als Vorlage dienen, deren Helden nur einen Angriffswert besitzen, indem sie für die einzelnen Waffen, die ein Charakter trägt, den gleichen Wert einsetzen.

Im *Szenarion* sind zum Einsatz von Waffen die folgenden drei Schlüsselworte im Charakter-File auszufüllen:

- **Nahkampfwaffe** : Hier wird die vom Charakter aktuell geführte Nahkampfwaffe aufgeführt. Darauf folgt der zugehörige Fähigkeitswert (s.o.). Der Meister trägt selbst die Verantwortung dafür, daß die ausgewählte Waffe auch nahkampftauglich ist. Spielercharaktere können während der Simulation eine neue Waffe aus ihrem Besitz nehmen. Kämpft ein Charakter waffenlos, so ist als Waffe die Faust einzutragen, bei “tierischen” Gegnern Waffen, die der Angriffsart in etwa entsprechen (z.B. Biß oder Stachel...).

Von den folgenden Waffen kann eine ausgewählt werden: {Dolch, Kurzschwert, Langschwert, Peitsche, Streitaxt, Keule, Kriegshammer, Morgenstern, Netz, Streitkolben, Kriegsflegel, Hellebarde, Lanze, Bihänder, Kampfstab, Speer, Wurfspeer, Schlachtbeil, Biß, Horn, Hauer, Stachel, Kralle, Klaue, Zange,

Schnabel, Kleinsttiere, Rammen, Zuschlagen, Stampfen, Trampeln, Umklammern, Packen, Umhüllen, Verschlingen, Erdrücken, Faust, Tritt, Ringen, Festhalten }.

Typ 3 - Syntax

- **Fernkampfwaffe** : Dieses Schlüsselwort gibt die zum Fernkampf geführte Waffe an. Darauf folgt der zugehörige Fähigkeitswert (s.o.). Neben klassischen Fernkampfwaffen können auch einige Nahkampfwaffen auf Distanz eingesetzt werden. Auch hier obliegt es der Sorgfaltspflicht des Meisters, nur fernkampftaugliche Waffen einzutragen.

Von den folgenden Waffen kann eine ausgewählt werden: {Dolch(8m), Kurzsword(4m), Streitaxt(8m), Keule(6m), Kriegshammer(6m), Netz(6m), Wurfspieß(16m), Streitkolben(6m), Speer(10m), Armbrust(36m), Bola(14m), Kompositbogen(30m), Kurzbogen(24m), Langbogen(40m), Schleuder(16m) }.

Der Wert in runden Klammern gibt die Reichweite der jeweiligen Waffe an. Die maximale Fernkampffentfernung ist durch die maximale Sichtweite auf einen Radius von 20 Metern um den Charakter begrenzt.

Typ 3 - Syntax

- **Waffen** : Hier bietet das Daten-File genug Raum, um alle Waffen, die ein Charakter bei sich trägt, aufzuführen. Dabei ist es egal, ob es sich um Fernkampf- und/oder Nahkampfwaffen handelt. Für jede Waffe ist der zugehörige Fähigkeitswert mit anzugeben. In der Simulation können Spieler-Charaktere dann hieraus, falls gewünscht, eine neue Waffe wählen.

Typ 5 - Syntax

Es bleibt anzumerken, daß Waffen unterschiedlicher Klassen verschiedene Angriffs- und Schadenstabellen benutzen. So kann ein meisterlicher Kopftreffer mit einem Kriegshammer den gegnerischen Kopf samt Harnisch zertrümmern, ein gleichwertiger Streich mit einem Schlachtbeil jedoch den Kopf vom Rumpf trennen...

Wie schon vorher erwähnt, ist der Gebrauch von Zweihandwaffe und Schild zulässig, da die Möglichkeit besteht, den Schild auf den Rücken zu gurten oder am Arm zu befestigen. Der Meister sollte jedoch den Waffen-Talentwert eines solchen Charakters erniedrigen (-25).

Talente und Fähigkeiten

Talente beschreiben die Fähigkeit eines Charakters, bestimmte Aktionen erfolgreich durchzuführen. Im Folgenden werden die im *Szenarion* Anwendung findenden Talente und Fähigkeiten genauer beschrieben.

- **Schwimmen:** Charaktere, die Wasserfelder betreten wollen, müssen mindestens über einen Schwimm-Wert von 5 verfügen. Anderenfalls weigern sie sich (aus gutem Grund): Schlechte Schwimmer können leicht ertrinken und kommen nur langsam vorwärts. Weiterhin sind sie vor Angriffen kaum geschützt. Wasserwesen sollten hier einen Eintrag von mindesten 150 aufweisen! Im Maximalfall kann sich ein Schwimmer nur mit der Hälfte seiner gewählten Geschwindigkeit fortbewegen. Mit zunehmender Geschwindigkeit erhöht der Schwimmer auch das Risiko, unterzugehen!

Typ 2 - Syntax

- **Verbergen:** Will ein Charakter sich auf einem Feld verstecken, so wird dieses Talent herangezogen. Ein versteckter Charakter kann aus dem Hinterhalt angreifen und ist für eigene und gegnerische Charaktere schwerer wahrzunehmen.

Typ 2 - Syntax

- **Öffnen:** Türen und Tore müssen zum Durchqueren geöffnet werden. Sind sie unverschlossen, so verlangsamen sie nur die normale Bewegung. Verschlossene Türen (..) jedoch müssen durch eine erfolgreiche Probe auf "Öffnen" erst einmal entriegelt werden.

Typ 2 - Syntax

- **Sprüche:** Magier, Barden und Anemisten besitzen die Macht, Magie anzuwenden. Dieser Talentwert bestimmt die Fähigkeit, Zauberformeln und Spruchlisten zu benutzen.

Typ 2 - Syntax

- **Wahrnehmung:** Über die Wahrnehmung erhält ein Charakter Informationen über seine Umgebung. Im *Szenarion* kommt diesem Wert eine Schlüsselrolle zu, denn Charaktere, die ihre Umwelt schlecht wahrnehmen, sind im Kampf extrem benachteiligt (Auswahl der Gegner, Bewegung..)! Durch getragene Rüstungsteile kann die Wahrnehmung behindert werden (z.B. Helm). Sollte im eigenen Rollenspiel-System kein solcher Wert vorhanden sein, so ist er aus den Grundeigenschaften (etwa Intelligenz oder Intuition) eines Charakters zu bilden und mit dem Bonus der entsprechenden Eigenschaft (siehe Tabelle 7.6.1) anzugeben.

Typ 2 - Syntax

- **Körperentwicklung:** Dieser Wert bestimmt die Lebensenergie, die ein Charakter/Monster maximal besitzen kann. Um die Konvertierung eigener Charaktere zu erleichtern, werden im Folgenden die Körperentwicklung für Helden aus den einzelnen "Berufen" angegeben:

Stufe	Krieger	Dieb	Priester	Magier	Barde	Waldläufer
1.	35	27	17	17	17	32
2.	45	32	22	22	22	38
3.	60	37	27	27	27	44
4.	75	42	33	33	33	50
<i>n.</i>	$20 + n \cdot 15$	$22 + n \cdot 5$	$12 + n \cdot 5$	$12 + n \cdot 5$	$12 + n \cdot 5$	$26 + n \cdot 6$

Eigene Werte können hieraus abgeleitet werden.

Typ 2 - Syntax

- **Lebenspunkte:** Während die Körperentwicklung die maximale Lebensenergie angibt, zeigt dieser Wert den aktuellen Zustand (physische Verfassung) eines Charakters an. Sinken die Lebenspunkte im Verlauf eines Kampfes auf Null, so stirbt der Charakter und wird aus der Simulation entfernt. Dieser Wert wird während der Kampfschauplatz - Simulation auch dargestellt und gibt so dem Spieler die nötige Information.

Typ 2 - Syntax

- **Magiepunkte:** Wie die Lebenspunkte die physische Verfassung widerspiegeln, so bezieht sich dieses Schlüsselwort auf die magische Aura, die ein Charakter besitzt. Sinkt dieser Wert nach Anwendung von Zaubern auf Null, so kann keine Magie mehr eingesetzt werden. Die maximale Höhe dieses Wertes richtet sich nach den Kosten für die Anwendung magischer Sprüche oder Gegenstände, wobei magische Gegenstände im *Szenarion* die Aura des Trägers verstärken, d.h. sie verleihen die Magiepunkte, die zur Anwendung nötig wären, ihrem Benutzer, der sie auch zur Anwendung anderer Magie anwenden kann. Dadurch werden auch Kieger in die Lage versetzt, Magie einzusetzen (allerdings nur Zauber, die sie aufgrund magischer Utensilien besitzen). Will der Meister die Magie seines Spielsystems an die vom *Szenarion* unterstützte anpassen, so kann folgende Tabelle zur Berechnung der Magiepunkte eines Charakters herangezogen werden.

Intelligenz	Magiepunkte
0-75	0
76-95	1
96-99	2
100-101	3
102+	4

Der aus der Tabelle abgelesene Wert wird dann mit der Stufe des Charakters multipliziert. Für Zaubersprüche gilt: Ein Zauber der 1.Stufe kostet einen Ma-

giepunkt, (..), ein Zauber der n .Stufe n Magiepunkte. Für Priester kann statt der Intelligenz auch eine Eigenschaft zur Berechnung der Magiepunkte gewählt werden, die die Verbundenheit zu ihrer Gottheit ausdrückt (z.B. Intuition).

Siehe auch Zauber und magische Formeln

Typ 2 - Syntax

- **Auftreten:** Dieser Wert soll das Charisma, also die Wirkung des Charakters auf andere, zum Ausdruck bringen. Er findet bei Gesprächen Anwendung, die zwei Charaktere miteinander führen. So kann ein PC mit sehr hohem Wert in "Auftreten" leicht einen anderen mit schwachem Wert überzeugen, die Partei zu wechseln und für ihn zu kämpfen oder Informationen preiszugeben.

Typ 2 - Syntax

- **Defensivbonus:** Er spiegelt die Fähigkeit eines Charakters wieder, Angriffe zu parieren. Als Grundlage zur Berechnung dieses Wertes dient zum einen die Eigenschaft Geschicklichkeit (der Bonus aus Tabelle 7.6.1), der Schild, der je nach Verarbeitung mit +20 bis +30 bewertet werden kann, sowie andere Einflüsse (z.B. Magie, spezielle Rüstungsteile). Im Kampf wird der Defensivbonus von dem Angriffswurf des Gegners abgezogen und die Auswirkungen für den Verteidiger so gemildert. Der DB steht zu Beginn jeder Simulationsrunde wieder voll zur Verfügung und wird durch jeden Angriff erniedrigt.

Typ 2 - Syntax

- **WWMagie:** Wie der Defensivbonus der Abwehr von Angriffen gilt, gilt der WWMagie-Bonus der Abwehr fremder Zauber. Eine Probe wird immer dann fällig, wenn auf einen Charakter ein Zauberspruch angewandt wird oder der Charakter sich an einen Ort begibt, an dem Magie wirksam ist. Zur Berechnung dieses Wertes können die Eigenschaften Intelligenz (z.B. Magier) und/oder Intuition (z.B. Priester) herangezogen werden, bzw. deren Boni durch Tabelle 7.6.1.

Typ 2 - Syntax

Ausstattung und Wissen:

Hier bietet das Daten-File die Möglichkeit, Zauberformeln und Informationen abzuspeichern, die ein Charakter im Kampf einsetzen kann oder die ihm ein persönliches Profil verleihen...

- **Spruchlisten:** Hier sind alle Zauber aufzuführen, die ein Charakter aufgrund seines Berufes (z.B. Magier oder Priester), seiner magischen Utensilien (z.B. Schwert mit eingebettetem Zauber..) oder durch die Zugehörigkeit zu einer bestimmten Rasse (z.B. Drache mit Feuerodem) in der Simulation einsetzen kann.

ACHTUNG: NPC's können bisher nur den ersten hier aufgeführten Spruch im Kampf benutzen. Daher sollte der Meister sich gut überlegen, welchen Spruch er hier z.B. für seine Monster einträgt!

Siehe auch Zauber und magische Formeln

Typ 4 - Syntax

- **Gegenstände:** Obwohl unter diesem Schlüsselwort schon Einträge vorgenommen werden können, ist es im *Szenarion* noch nicht implementiert. Dies wird ggf. zu einem späteren Zeitpunkt nachgeholt.

Typ 4 - Syntax

- **Wissen:** Diese Information teilt ein Charakter einem anderen mit, der ihn erfolgreich während der Simulation ausfragt. Wie gut er sie hütet, bzw. erfragen kann, hängt vom Wert Auftreten (siehe dort) ab.

Typ 1 - Syntax

- **Kampfschrei:** Ist ein Charakter verzweifelt und von vielen Gegnern umgeben, so kann er einen Rundumschlag unternehmen. Er sammelt dann all seinen Mut

und seine Kraft zusammen, schreit den hier angegebenen Spruch seinen Feinden entgegen und unternimmt einen gewaltigen Streich, während er sich um seine eigene Achse dreht.

Typ 1 - Syntax

7.7 Zauber und magische Formeln

Das Magiesystem des *Szenarions* ist bewußt offen gestaltet, um allen Meistern die Möglichkeit zu geben, die Zaubersprüche ihres Spielsystems einzubetten. Natürlich ist auch hierbei wieder eine Normierung durchzuführen.

Allgemein kann man sagen, Magie ist im *Szenarion* ortsbezogen. D.h., ein Magier spricht einen Zauber nicht auf eine Person aus, sondern auf einen Ort, auf dem dann, je nach Wirkungsdauer, der Zauber aktiv ist. Kreaturen, die dieses Feld betreten, erleiden Schaden (falls ein Rettungswurf nicht gelingt) oder genießen die Vorzüge (z.B. Unsichtbarkeit).

7.7.1 Anwendung von Zaubern

Zaubern kann jeder Charakter, der Einträge in seinem Daten-File unter dem Schlüsselwort *Spruchlisten* hat. Dies können Krieger mit magischen Utensilien oder Magier (..) sein. Wichtig ist nur, daß noch genügend Magiepunkte (siehe 7.6.3 Magiepunkte) vorhanden sind, der Anwender also noch die magische Aura besitzt, die zur Erschaffung eines Zaubers²⁷ nötig ist.

Wählt ein Charakter das *Zaubern* als Aktion in der Simulation, so wird eine Probe auf den Charakter-Wert *Sprüche* fällig. Ist diese Probe erfolgreich, so werden die betroffenen Felder ermittelt und mit dem Zauber für eine gewisse Zeit (s.u.) "belegt". Danach erst (und dann in jeder folgenden Runde) wird geprüft, ob auf den Feldern Charaktere anwesend sind, denen dann jede Runde, auf der sie auf einem "verzauberten" Feld stehen, ein Widerstandswurf (*WWMagie*) zusteht. Mißlingt diese Probe, so erleiden sie den Schaden, der durch die Klasse des Spruches impliziert wird. Dabei spielt die Stufe desjenigen, der den Spruch erzeugt hat und demjenigen, der vom Zauber betroffen ist, sowie die Stärke des Zaubers selbst eine wichtige Rolle.

²⁷Die Kosten eines Zauberspruches muß der Meister anhand der Mächtigkeit berechnen!

Heilzauber bilden eine Ausnahme, da sich ein Ziel kaum gegen die Zuführung von Lebenspunkten wehren wird.

Felder, auf denen Schadenszauber präsent sind, behindern die Bewegung. Hitze, Kälte und andere extreme Auswirkungen reduzieren die Geschwindigkeit der Charaktere auf ein Feld pro Runde.

Die Wirkzeit von Zaubern ist durch die *Dauer* festgelegt. Erlittener Schaden (z.B. Lebenspunkte) jedoch wird auch nach dem Ablauf eines Zaubers nicht wieder rückgängig gemacht.

7.7.2 Arten von Zauberei

Die Zauber im *Szenarion* sind in Klassen zusammengefaßt. Die Zaubersprüche einer Klasse ähneln sich in ihrer Wirkung auf "Opfer". Neue, vom Meister erzeugte Zauber sind zuerst einer der folgenden Klassen zuzuordnen, sofern sie in irgend einer Weise auf den Ausgang eines Kampfes Auswirkungen haben können:

- **Lebenspunkte:** Hierunter fallen alle Zauber, die zur Regeneration oder Heilung der Lebenspunkte des eigenen oder fremder Charaktere angewandt werden können.
- **Schaden:** Diese Zauber fügen einem Opfer Schaden zu. D.h., die Magie verringert die Lebenspunkte eines Opfers (z.B. durch Feuerodem oder Kälteschock).
- **Wahrnehmung:** Andere können einen durch solch einen Zauber geschützten Charakter schlechter wahrnehmen. So können Zauber dieser Klasse Nebel erzeugen oder gar unsichtbar machen...
- **Geschwindigkeit:** Betroffene Charaktere haben eine erhöhte oder erniedrigte Reichweite (z.B. durch Lähmung oder Beschleunigung).
- **Gegenzauber:** Diese Sprüche dienen zur Eindämmung/Verstärkung fremder oder eigener Magie. Die Rettungswürfe betroffener Kreaturen erhöhen/erniedrigen sich. Typische Vertreter dieser Klasse sind Schutzzauber aller Art. Ist

im Zentrum eines Gegenzaubers bei der Aussprache des Spruches ein anderer Zauber anwesend, so kann der Gegenzauber den anderen Zauber samt Wirkung aufheben (Nicht jedoch bei Metamorphose-Zaubern).

- **Benommen(heit):** Als benommen zählen alle Charaktere, die in einer Simulationsrunde nicht aktiv am Geschehen teilnehmen und so eine eigene Handlung wählen können. Dabei ist es unerheblich, ob sie sich im Schlaf befinden oder bewußtlos geschlagen wurden.
- **Öffnen:** Durch das PlantFlag *close* lassen sich Türen, Tore, Fenster (..) verschließen. Zauber dieser Klasse können diese Wirkung aufheben.
- **Verschließen:** Zauber dieser Klasse können alle Felder, die das PlantFlag *open* tragen, unpassierbar machen.
- **Terraform:** Zielt ein Zauber auf die Veränderung der Landschaft, so ist er in diese Klasse Einzuordnen (so z.B. die Erschaffung eines Berges). Charaktere, die sich im Wirkungsgebiet befinden, erleiden Schaden.
- **Furcht:** Sprüche, die ihre Gegner einschüchtern, werden hier zusammengefaßt. Die Opfer solcher Magie kämpfen schlechter und unkonzentrierter.
- **Teleport:** Die Zaubersprüche dieser Klasse dienen zur Überbrückung von Entfernungen. Unpassierbare Täler (..) können so überwunden werden.
- **Metamorphose:** Soll ein Opfer in eine andere Gestalt (z.B. in einen Frosch) verwandelt werden, so ist ein Spruch aus dieser Klasse zu benutzen.

Es kann nötig sein, einen Zauber aus dem eigenen Spielsystem in mehrere Szenarion-Zauber zu übersetzen, falls die Wirkung nicht eindeutig mit den hier beschriebenen Mitteln definiert werden kann.

7.7.3 Syntax von Zaubern

Von der Syntax her gleicht der Aufbau eines Spruches aus dem *spells/* Verzeichnis einem Charakter-Daten-File (siehe Syntax Typen aus 7.6.2). Auch hier gelten selbige Restriktionen.

Folgende Schlüsselworte sind zur Erzeugung eines Zauberspruches auszufüllen:

- **Name:** Unter den hier aufgeführten Namen muß der Zauber im Verzeichnis *spells/* mit Anhang “.spell” eingetragen werden.

Typ 1 - Syntax

- **Wirkung:** Die unter 7.7.2 aufgeführten Arten der Zauberei sind als Eintrag zulässig (siehe dort). Somit ordnet dieser Schlüssel den Spruch einer bestimmten Klasse zu.

Typ 1 - Syntax

- **Radius:** Hier wird festgelegt, in welchem Radius in Feldern ein Zauber wirken soll. Dabei ist bei Sprüchen, die nur auf ein Feld (ein Ziel) wirken dürfen, eine Null (0) einzutragen. Bei einer Eins als Eintrag kommen alle Charaktere in den “Genuß” des Spruches, die bis zu einem Feld entfernt vom Ziel stehen (..), der Anwender des Spruches jedoch nicht mehr!

Typ 2 - Syntax

- **Reichweite:** Sie bezieht sich auf die Entfernung bis zum Ziel in Feldern. Der Wirkungsbereich selbst wird durch den *Radius* festgelegt. Als Ziel können Gegenstände oder Charaktere gewählt werden. Soll nur der Erzeuger des Zaubers von der Wirkung betroffen werden, so ist hier und unter dem Schlüsselwort *Radius* eine Null (0) einzutragen.

Typ 2 - Syntax

- **Dauer:** Dieser Wert gibt die Wirkzeit eines Zaubers in Runden an. Als Mindestdauer ist hier eine Eins anzugeben. In diesem Fall wirkt der Spruch für genau die Runde, in der er ausgesprochen wird.

Typ 2 - Syntax

- **Wert:** Das Schlüsselwort “Wert” hat nicht in jeder Klasse Bedeutung. In der Klasse *Schaden & Terraform* bestimmt es die Schadensart: Hitzeschaden (6), Kälteschaden (7) oder Schaden durch Elektrizität (8), sowie magischer Hieb- (1), Streich- (2) und Stichschaden (3) sind zulässig.

Wahrnehmung: Hier bestimmt der “Wert” den Malus, den fremde Charaktere bei einem Wahrnehmungsversuch erhalten. Außerdem wird der Angriffs- und Verteidigungswert damit modifiziert (jeweils $\frac{Wert}{2}$).

Geschwindigkeit: Der Wert wird explizit als Reichweite in Feldern für die nächste Bewegung angesehen.

Ein *Gegenzauber* erhöht den Widerstand gegen Magie um den angegebenen Wert.

Typ 2 - Syntax

- **Verstärkung:** Der hier angegebene Wert erhöht die Wirkungsdauer in Abhängigkeit von der Stufe des Zaubers. Soll dies nicht geschehen, so ist eine Null (0) einzutragen, andernfalls der Faktor, mit dem die Wirkzeit pro Stufe multipliziert werden soll (ganzzahlig).

Typ 2 - Syntax

- **Stärke:** Dieser Wert erhöht bzw. erniedrigt die Wahrscheinlichkeit des Gelingens eines Spruches. Für Zaubersprüche der Klasse *Geschwindigkeit* gibt er den Offensivbonus an, für die Klasse *Lebenspunkte* die Anzahl der zu regenerierenden Lebensenergie.

Typ 2 - Syntax

- **Opfer:** Dem Opfer bzw. Ziel eines Zauberspruches wird der hier stehende Text als Erklärung für das mitgeteilt, was es spürt, hört und sieht.

Typ 1 - Syntax

- **AT:** Hier wird nochmals die Art des Zaubers unterschieden. Angriffssprüche müssen eine "9", die übrigen eine "10" als Eintrag bekommen.

Typ 2 - Syntax

- **Kosten:** Dieser Wert gibt die Anzahl der Magiepunkte an, die zur Erzeugung des Zaubers notwendig sind. Als Richtwert sollte gelten: Ein Spruch des 1. Levels kostet auch einen Magiepunkt...

Typ 2 - Syntax

- **Spezial:** Metamorphosezauber: Das Opfer bzw. Ziel eines Zaubers wird in den hier referenzierten Charakter aus dem *characters/*Verzeichnis verwandelt. Bei Sprüchen aus der Klasse Terraform ist eines der folgenden Worte einzutragen: **Land** (Erschafft auf Wasser Land), **Berg** (ein schlecht passierbarer Berg entsteht), **Hügel** (ein die Sicht behindernder Hügel formt sich) oder **Tal** (ein Senke bildet sich aus).

Typ 1 - Syntax

7.7.4 Beispiel zur Sprucherzeugung

Besuchen wir wieder unseren imaginären Meister. Diesmal will er es seinen Spielern wirklich schwer machen: Er hat einen mächtigen Gegner erzeugt, einen Drachen. Dieser verfügt nicht nur über sehr gute physische Werte, sondern soll auch mittels Feuerodem seine aufdringlichen Helden in Schach halten können.

Zuerst ordnet der Meister den Spruch in eine Klasse zu: Klar, Drachenodem stinkt und schmerzt, daher wählt er als Wirkung den *Schaden*.

Weiter denkt sich unser Meister, sein Drache hat eine enorme Lunge und kann so eine Unmenge von Flammen (Radius 3 = 12 Meter Durchmesser) sehr weit (10

Felder = 20 Meter) spucken. Die Flammen erzeugen einen schweren (Stärke +20) Hitzeschaden (Wert 6).

Die nach Schwefel stinkenden Indrigenzien des Odems verflüchtigen sich schnell, daher erlischt das Flammenmeer nach einer Runde (Dauer 1).

Das fertige File speichert der Meister im Verzeichnis *spells/* unter dem Namen "Feuerodem.spell" ab.

```
Name,Feuerodem
Wirkung,Schaden
Radius,3
Reichweite,10
Dauer,1
Wert,6
Verstärkung,0
Stärke,+20
Opfer,Du_tauchst_in_ein_riesiges_Meer_von_Flammen_ein.
AT,9
Kosten,5
Spezial
```

Vorher hatte er natürlich den Priester der Gruppe, Gronar, mit einem Heilzauber versehen. Der Heilzauber des Orginalsystems war sowohl für den Erzeuger, als auch für andere Charaktere gedacht. Daher teilte der Meister ihn in zwei Sprüche auf, einen zur Selbstheilung (SelbstheilenI.spell) und einen zur Heilung anderer (FremdheilenI.spell).

Weitere Zaubersprüche finden sie im Anhang.

8 Anpassung eigener Spielsysteme

In seinem Leben als Rollenspieler lernt man eine Vielzahl verschiedener Spielsysteme kennen, lieben und hassen. Fast jeder Rollenspieler hat, zumindest im Hinterkopf, einige Verbesserungsvorschläge, selbst für sein Lieblingssystem. Warum sollte man nun gerade das *Szenarion* benutzen, zumal es noch mit einem lärmenden Rechner verbunden ist, der die Fantasy-Stimmung gegen Null konvergieren läßt?

Nun, ich kann dies nur aus meiner eigenen Perspektive begründen. Ich versuche, meine Spieler nicht mit Regelwerk zu erdrücken, niemand, außer mir, muß die Regeln kennen. Dadurch wurden zum einen nervenaufreibende Regelauslegungen und der damit verbundene Streit zwischen Meister und Spieler vermieden und ich gewann Zeit zum Improvisieren²⁸, zum anderen erforderte dies von mir in Kämpfen sechs Arme und zwölf Augen, damit ich in angemessener Zeit Tabellen nachschlagen, Informationen geben und Modifikationen ausrechnen konnte. Da ich kein echter Multitasking-Meister²⁹ bin, haben mich Kämpfe oft den letzten Nerv gekostet. Bei mir und nicht bei meinen Spielern sammelten sich jedesmal Schweißperlen auf der Stirn, das konnte einfach nicht richtig sein — Auf dieser Basis entstand das *Szenarion*!

8.1 Was bleibt, was ändert sich?

Gleich vorweg, das eigene Spielsystem bleibt, bis auf Kämpfe, für alles andere zuständig. Man bewegt sich, klettert, fällt, lebt und stirbt wie gehabt. Kein Spieler muß also auf seine geliebten Würfel verzichten!

Allein die Simulation der Kämpfe erfordern vom Meister einen gesunden Menschenverstand, um die Werte seines Rollenspielsystems in die des *Szenarions* zu

²⁸Dies senkte auch spürbar die Vorbereitungszeit für ein (selbstgeschriebenes) Abenteuer

²⁹Ich kann höchstens Prozesse schlafen legen wie ein Mac!

übersetzen (normieren). Dazu ist es einmalig nötig, die Charakter-Werte zu übersetzen und die Zaubersprüche der Charaktere bereitzustellen (siehe in den entsprechenden Kapiteln).

Natürlich ändern sich auch die aus einem Kampf entstehenden Verletzungen. Die Simulation verwendet ihre eigenen Tabellen an, die getrennt nach den verschiedenen Waffen benutzt werden und auch eigene kritische Treffer erzeugen. D.h., man muß sich an das neue Kampfsystem erst einmal gewöhnen.

8.2 Fazit

Für einen erfahrenen Rollenspieler dürfte es kein Problem sein, die oben erwähnten Normierungen durchzuführen. Es bleibt auch ihm überlassen, ob er das *Szenarion* oft, immer oder gar nicht benutzt. Wer bisher in Kämpfen immer Herr der Lage war, Spannung erzeugen konnte und dabei auch selbst noch Spaß hatte, der hat dieses Hilfs-Tool natürlich nicht nötig. Wer jedoch kein wandelndes Regelbuch ist, eigene Ideen in den Vordergrund stellt und die gewonnene Zeit zur besseren Ausschmückung des gesamten Abenteuers oder zur Improvisation nutzen möchte, wird vom *Szenarion* nicht im Stich gelassen!

Letztlich und endlich ist der Kampf nur ein Element von vielen, die zu einem gelungenen Rollenspiel beitragen — auf die richtige Dosierung kommt es an. Dafür trägt der Meister die alleinige Verantwortung!

9 Szenarion Multi-User-Interface

Hinter diesem Namen verbirgt sich nichts anderes als ein Joystick, dessen vier Bewegungsrichtungs-Kontakte aus dem Gehäuse herausgeführt werden müssen.

ACHTUNG: Unsachgemäße Durchführung der hier beschriebenen Arbeitsschritte kann zu Beschädigungen des Rechners führen. Vor dem Beginn der Arbeit immer alle Verbindungen des Joysticks zum Rechner lösen und statische Aufladungen des Interfaces vermeiden. Ich selbst habe mir durch Dummheit beim Löten bei nicht gezogener Verbindung ein IC zerschossen, was mich Zeit und Nerven gekostet hat. Immerhin, auf dem Weg dahin ist jetzt alles gesockelt :)

Für entstehende Schäden besteht keinerlei Garantie-Anspruch! Die Ausführung der hier beschriebenen Arbeitsschritte geschieht auf eigene Verantwortung!

Multi-User-Interface

Zum Bau des Interfaces benötigen sie einen (alten) Joystick, ca. 8 Meter Kabel, sowie 4 Druckknöpfe, also Material für weniger als 20 DM! Verwenden sie kein Potentiometer, da hier andere Pins als bei einem Joystick verwendet werden!

- Öffnen sie zuerst den Joystick und nehmen sie den Steuerhebel heraus. Er wird nicht mehr benötigt.
- Ersetzen sie die vier Kontakte, die durch den Hebel geschlossen wurden, durch je 2 Kabel, die sie statt der Kontakte an den entsprechenden Leiterbahnen der Platine anlöten. Die insgesamt 8 Kabel müssen nach außen geführt werden. Danach kann das Joystick-Gehäuse wieder geschlossen werden.
- Nun die jeweils zwei zusammengehörenden Kabel (DGND und je eine Bewegungsrichtung) durch einen Druckknopf verbinden (löten). Um eine bessere Handhabbarkeit zu erzielen, sollten die Druckknöpfe auf je ein kleines Brett, ein eigenes Gehäuse o.Ä befestigt werden. Ich selbst habe die Kabel durch je eine Röhre geführt, dann die "Feuerknöpfe" mit den Kabeln verbunden und auf dem Ende der Röhre befestigt.

Wenn keine Fehler aufgetreten sind, kann jetzt mit jeder isolierten Bewegungsrichtung in einem Szenario mit vier Spielercharakteren ein Spieler-Fenster gesteuert werden.

10 Kontakt

Grundsätzlich freue ich mich über Anregungen, Kritik und Verbesserungsvorschläge, sowie Bug-Reports. Wer also mit mir in Kontakt treten möchte, kann dies entweder über meine E-Mail Adresse:

stefanw@ida.ing.tu-bs.de

oder privat:

**Stefan Waschk
Westerberg 9
D-38173 Sickinge**

Postalischen Fragen ist ein frankierter Rückumschlag beizulegen. Bestellungen der Vollversion nehme ich nur gegen Vorkasse durch Überweisung auf das unten genannte Konto (36.90 DM im Inland, davon 6.90 DM Porto, 42 DM für Bestellungen aus dem Ausland, incl. 12 DM Porto) und zusätzliche schriftliche Bestellung an. Im Lieferpaket enthalten sind:

- zwei Installationsdisketten (DD)
- ein gebundenes Regelheft

Meine Bankverbindung:

**Volksbank Braunschweig
BLZ: 270.900.77
K.Nr: 337.928.006**

11 Bugs & Fehler

An erster Stelle müssen hier die Schnittstellen von Tools zum Meister genannt werden. Aufgrund von Eingabefehlern in vom Meister erzeugten Daten-Files kann es zu unkontrollierten Programmabstürzen kommen, da die vorgeschalteten Parser nicht alles abfangen. Es bleibt abzuwarten, ob sich eine für die Öffentlichkeit gedachte Weiterentwicklung des *Szenarions* lohnt, da ich selbst keine Schwierigkeiten habe, die von mir erdachten Regeln einzuhalten. Meine Priorität liegt daher eher in der Weiterentwicklung der Simulation (einfacheres setzen von Feldflags, Benutzen, Reiten, englische Texte, Gouraud-Shading (Phong(?)< – >Zeit(!) etc.).

Leider können tote Charaktere können nicht mehr mit dem Meister-Menu bearbeitet werden. Auch zeigen manche Sprüche keine Wirkung, die mit dem Meister-Menu erzeugt wurden.

Vor dem Setzen von Verbindungen unbedingt abspeichern (Absturz möglich!!!)

Die Wahl eines geeigneten Screenmodes für A4000/A1200 konnte noch nicht zufriedenstellend realisiert werden. Es bleibt abzuwarten, welche Screenmodes in Zukunft für den Amiga zur Verfügung stehen werden... Ich selbst würde mir eine Auflösung von 1024x768 (benutze ich z.Zt) oder 1248x1024 (bis zu 20 Spieler-Charaktere!!!) in 256 Farben wünschen... (von RTG!!!), Resource-Tracking, Memory Protection... ganz zu schweigen) Leider werden Grafikkarten nicht unterstützt, da WritePixel, ClipBlit (..) einfach ein zu schlechtes Laufzeitverhalten zeigten, hoffen wir also alle auf RTG...

12 Anhang

12.1 Szenarion Zauber

Hier nun eine alphabetische Auflistung der mitgelieferten Zaubersprüche. Sie sollen als Vorlage für das eigene Magiesystem dienen. Diese Liste erhebt keinen Anspruch auf Vollständigkeit. Der Meister muß sich selbst Gedanken machen, welche Zauber er für sein Magiesystem benötigt.

Berg_erschaffen: Ein Berg wird erschaffen (Spezial = Berg, Durchmesser 8 Meter). Charaktere, die auf den betroffenen Feldern stehen, erleiden einen Hieb-schaden (Wert = 1) durch Stürze.

Benommenheit_I: Dieser Zauber macht sein Opfer benommen, falls der ein normaler Rettungs mißlingt (Stärke = 0). Der Spruch bleibt $5 \cdot \text{StufedesZauberers}$ Kampfrunden lang aktiv und wirkt nur auf dem Zielfeld (Radius = 0).

Feueraxt: Dieser Zauber ist in Gronars (*siehe characters/Gronar.char*) magische Axt eingebettet. Er fügt für 2 Kampfrunden dem Opfern, das dem Spruch erliegt (Radius = 0), einen Hitzetreffer zu (Wert = 6, Hitzeschaden). Der Spruch wirkt mit recht hoher Wahrscheinlichkeit (Stärke = 50).

Feuerodem: Wie auch bei dem Zauber Feueraxt erleiden die Opfer, die sich hier innerhalb eines Radius von 6 Metern aufhalten, einen Hitze Hitzeschaden. Als Caster ist ein Drache vorgesehen. Das Zentrum der Spruchs kann bis zu 20 Meter entfernt liegen (*Reichweite* · 2).

Flammenruf: Auch dieser Zauber erzeugt einen Hitzeschaden, ist aber nicht ganz so mächtig wie der vorher beschriebene Feuerodem (vergleiche dort).

Flächenschutz: Dieser Zauber schützt einen Umkreis von 4 Metern um den Anwender herum für *Stufe* · 3 Runden. Außerdem kann er auf dem Feld des Anwenders befindliche Sprüche aufheben.

Frosch_I: Opfer, die bis zu 20 Metern entfernt stehen können, verwandeln sich in einen Frosch (characters/Frosch.char). Die Wirkung auf die Opfer ist permanent!

Geschwindigkeit_I: Ein Zauberer kann einem bis zu 4 Metern entfernt stehenden Charakter eine Reichweite von 11 Feldern für die nächste Bewegungsrunde verleihen.

Halten_I: Das bis zu 10 Metern entfernt stehende Opfer kann in der nächsten Bewegungsrunde nur noch ein Feld weit ziehen. Der Spruch bleibt auf dem Zielfeld *Stufe* · 4 Runden lang bestehen.

Heilen_Anderer_I: Der Caster dieses Spruches kann einem anderen Charakter, der sich auf einem angrenzenden Feld aufhalten muß, 6 Lebenspunkte zuführen.

Heilen_Selbst_I: Der Zauberer kann sich mit diesem Spruch selbst neue Lebensenergie zuführen (6 Lebenspunkte).

Kältewand_I: Im Gegensatz zu Schadenszaubern wie Flammenruf, erzeugt dieser Kälteschaden auf einem bis zu 20 Metern entfernten Feld. Die Wirkung hält *Stufe* · 10 Runden lang an.

Nebel_I: Die Wahrnehmung von Charakteren, sowie für Außenstehende, die in dieses Gebiet (Durchmesser 12 Meter) sehen, wird erschwert. Der Anwender kann sich so Fernkampfangriffen entziehen. Die Wirkung hält *Stufe* · 3 Runden lang an.

Öffnen_I: Der Anwender dieses Zaubers öffnet alle Schlösser, die sich auf angrenzenden Feldern befinden.

Schloß_I: Der Anwender dieses Zaubers verschließt alle Schlösser, die sich auf angrenzenden Feldern befinden.

Teleport_I: Alle Charaktere innerhalb von 4 Metern Entfernung vom Caster und er selbst teleportieren sich zu einem Ziel, daß bis zu 30 Metern entfernt liegen kann.

Verhüllen_I: Der Anwender dieses Zaubers verschmilzt mit der Umwelt und wird so für alle anderen Charaktere extrem schlecht wahrnehmbar — er ist somit fast unsichtbar (-40 auf Wahrnehmung für alle anderen).

12.2 Babyschritte...

So, nun genug der vielen Regeln. Lassen sie sich bitte davon nicht abschrecken! Das nun folgende Beispiel soll Sie und Ihre Spieler Schritt für Schritt im Umgang mit dem Szenarion vertraut machen. Sobald sie das **Szenarion installiert** haben, kann es auch schon losgehen!

Starten:

Wechseln Sie in das Verzeichnis `../Szenation/` und geben in die *AMIGA-Shell*, *CShell* o.ä. (mit einer mindest-Stack-Größe von 250000 Byte) den Startbefehl für das Simulationstool RKSIm ein:

```
9> RKSIm
```

und drücken sie *ENTER* oder starten sie dieses Tool einfach durch einen Doppelklick des *Icons* . Nach kurzer Zeit erscheint ein *File-Requester*. Wählen sie die Simulation *Demo.map*.

Die Voreinstellungen:

Der RKSIm öffnet seinen eigenen Screen (je nach Einstellung im ENV/Szenarion.config) und bietet dem Meister nach dem Ladevorgang und vor der ersten Simulationsrunde die Möglichkeit, das Szenario zu modifizieren. Folgende Voreinstellung für das Demo-Szenario wurde gewählt:

- ein Spieler (Morgai, der Waldelbe), für ihn gelten die Informationen, die im Charakterfenster angezeigt werden. Die Steuerung erfolgt entweder über die Tastatur mit den Tasten **Q** (nächsten Menu-Punkt anwählen) und **A** (Menu-Punkt auswählen), das Multi-User-Interface, oder einen Joystick in Port 2 mittels Bewegungsrichtung UP
- zwei Freude des Elben, Gronar, der Zwerg und Hawk, der Dieb. Sie gehorchen Morgai (siehe Verhaltensflag "obeye")

- Freunde und Feinde, die auf dem Kampfschauplatz darauf warten, entdeckt zu werden...(siehe *allfile/demo.alf*)

Mit einem Mouseklick auf *DONE* bestätigt der Meister die Voreinstellung. Im folgenden repräsentieren **Meister** die Informationen, die der Elb vom Meister, also in diesem Fall vom RKSIm in seinem Fenster bekommt und **Morgai** die Aktionen, die der Spieler ausführt.

Der Wachturm am Fluß:

Achtung: Der im folgenden beschriebene Ablauf kann beim Nachspielen variieren, da die Texte und Proben zufällig erzeugt werden. Viel Spaß beim Nachspielen!

Meister an Morgai: Auf einem Hügel in der Nähe liegt ein alter Wachturm. Der Meister führt einen Wahrnehmungswurf für den Elben aus, der mittelmäßig ausfällt: Er macht einen bewohnten Eindruck. Du stehst und befindest Dich in einer Senke in hügeligem Gelände. Nahe bei sind 2 Buchen. Du blickst Dich um: Da, im Süd-Osten bei Dir, ein Corsar: Es ist Hawk. Hawk steht gen Norden. Der Pfeil im Fenster zeigte kurz auf ihn, jetzt aber auf den Zwerg, da er der letzte ist, der in diesem Absatz mit (s.o.) aufgeführt wurde. Da, im Osten neben Dir, ein Zwerg: Es ist Gronar. Gronar steht gen Norden. Du bemerkst im Süd-Osten einen Dunedain, bestimmt 10 Meter entfernt. Der Dunedain steht mit Dolch gen Norden. Jetzt zeigt der Pfeil auf den Dunedain. Lege nun Dein Schicksal fest!

Morgai: Er weiß, daß er mit seinem Langbogen auf diese Distance gut treffen kann. Da seine Gefährten wahrscheinlich auf diesen Dunedain zugehen werden, beschließt er, sich den Fremden genauer anzusehen. Es wird sich ja bald zeigen, wie er sich verhält. Nächste Runde kann er immernoch seine Freunde zum Einlenken bewegen. Der Elbe wählt *Sta/Sie/ Dunedain*

Meister: *Er schildert Morgai sofort die Informationen, die ihm wegen der gelungenen Wahrnehmungsprobe zustehen.* Du siehst im Nord-Osten einen Dunedain, gut 10 Meter entfernt. Ein Mann in lumpiger, abgerissener Kleidung. Er will Dir etwas mitteilen. Du siehst, der Dunedain trägt folgende Ausrüstung: Ohne_Rüstung, Dolch.

Situation: Hawk und Gronar sind kampfbereit auf den “freundlichen” Mann zugegangen und könnten ihn jetzt mit Fernkampf oder in der nächsten Runde mit Nahkampfangriffen verletzen. Der Elbe Morgai muß dies verhindern.

Meister an Morgai: *Zuerst wieder die allgemeine Beschreibung ...* Im Nord-Osten ist ein Corsar, gut 8 Meter entfernt. Es ist Hawk. Hawk steht gegen einen Dunedain. Du bemerkst im Nord-Osten einen Zwerg, ca. 6 Meter entfernt. Es ist Gronar. Gronar steht gegen einen Dunedain. Da, im Nord-Osten, ein Dunedain, um die 10 Meter entfernt, der Dunedain steht mit Dolch gen Norden. Lege nun Dein Schicksal fest.

Morgai: Er sieht, daß seine Freunde gleich den Mann erreicht haben. Schnell gibt er ihnen den Befehl *Sta/Bef/ Haltet ein!*

Meister: Du gebietest Gronar und Hawk, die Kampfhandlungen einzustellen.

Situation: Die beiden Mitstreiter des Elben haben den Dunedain erreicht, greifen ihn jedoch nicht an, da sie die Weisung ihres Anführers gehört haben. *Hätte einer von ihnen keinen Kontakt mehr zu Morgai gehabt, so hätte er seinen Angriff fortgesetzt!*

Meister an Morgai: *Zuerst wieder die allgemeine Beschreibung, Hawk und Gronar warten ab.*

Morgai: Er will mit dem Dunedain sprechen, daher läuft er schnell zu ihm, bevor seinen Freunden doch noch einfällt, anzugreifen! Seine Aktion: *Bew/Lau/Dunedain*

Situation: Noch halten sich Hawk und Gronar an ihren Befehl. Morgai läuft zum Dunedain. Dieser bewegt sich nicht.

Meister an Morgai: Du läufst und bist in ebener Umgebung. Du blickst Dich um (..) Da im Nord-Osten neben Dir, ein Dunedain. Der Dunedain steht mit Dolch gen Norden. Lege nun Dein Schicksal fest!

Morgai: Jetzt endlich steht er neben dem Fremden. Er spricht ihn an (*Sta/red/Dunedain*).

Meister: *Eine erfolgreiche Probe auf Morgais Talent, mit Menschen umzugehen und das gesetzte Flag join des Dunedain führen dazu, daß er seine Informationen (aus dem Allfile) verrät und sich selbst der Gruppe anschließt: Du sprichst einen Dunedain an. Der Dunedain erwidert: “ Ich bin der Seemann Herbert! Meine Freunde gingen in den Turm und kehrten nicht zurück!”. Lange blickt Ihr Euch an. Schließlich lächelt Dein Gesprächspartner und sagt: “Was Ihr sagt, Freund, überzeugt mich. Ich bin von nun an an Eurer Seite!”.*

Situation: Das kluge Verhalten des Elben hat der Gruppe einen weiteren Mitstreiter beschert. So können sie nun zu viert den Turm, der sich im Norden abzeichnet, erkunden. Dank Herberts Warnung tapen sie nun nicht in eine Falle! Später kann Morgai seine Gefolgsleute durch den Befehl *Sta/Bef/send* *Untergebene* zum Turm schicken. Vorher sollte er jedoch nicht vergessen, sie zur Vorsicht zu animieren (*Sta/Bef/ Greift ein!*).

Tip: Falls sie die Türen nicht öffnen können (z.B. durch *Bew/Ann/ Tür*, *Bew/Geh/ Tür ...*), schicken sie Hawk durch die Tür (*Sta/Bef/sen/ Tür*). Hawk hat als Dieb bessere Chancen, dieses Hindernis zu bewältigen! Wenn sie Im Haus nicht zuerst die Treppen hinab ins Unbekannte steigen wollen, können sie auch durch *Sta/Bew/Sen/ Stufen_abwärts* ihre Gefolgsleute vorausschicken!.

Viel Spaß bei der weiteren Erkundung des Turmes!

Im Verzeichnis *maps/* finden sie noch weitere Szenarien, die sie ebenfalls testen sollten:

KillingFields.map stellt dabei eine (von mir) nicht lösbare Aufgabe an den/die Spieler. Diese haben sich auf einen Hügel zurückgezogen und werden von über 40 Gegnern bedrängt. Die Anzahl der beteiligten Charaktere stellt auch eine Art bisheriger Leistungsgrenze des Szenarios dar. Größere Kämpfe grenzen schon eher an Schlachten, für die sich Spiele wie Table-Top besser eignen!

FieldsOfHonour.map stammt aus einem meiner Abenteuer. Die “Helden” wurden von einem dunklen Magier an diesen Ort des Grauens gegen ihren Willen teleportiert. Eigentlich hatte er auch versucht, sie mit einem Schlaf-Spruch bewußtlos zu machen. Zwar wirkte der Teleport, der sie in den Kerker des Magiers brachte, der Schlaf-Spruch aber konnte abgewendet werden, so daß sie eine reelle Chance haben, gegen die Wächter im Kerker zu gewinnen.

GuardTheKeep.map Die drei Charaktere Gronar, Morgai und Hawk, sowie sieben Soldaten, die den Freunden unterstellt sind, versuchen, eine alte Festung am Meer gegen anstürmende Insulaner zu verteidigen. Versuchen sie ihr Glück und spielen sie dieses Abenteuer als einer der Hauptcharaktere...

Mein Tip: Ich selbst habe während des Abenteuers die Rolle des Dunklelben übernommen. Meine Charaktere mußten 30 Runden überstehen, danach traf von der Seeseite ein imaginäres Schiff ein, auf das sie sich retten konnten (jeder, der den Strand erreicht hatte, wurde durch das Meister-Menu entfernt...

PlainField.map ist ein Experimental-Szenarion für den eiligen Meister. Wer keinen Wert auf das Aussehen des Kampfschauplatzes legt, kann hier in das entsprechende Allfile die beteiligten Charaktere unter Angabe der Feldkoordinaten eintragen und diesen “ebenen” Kampfschauplatz ohne RFEditor mit Leben füllen.