

DipWork

Version 2.0

Torsten Will

Melanchthonstr.87

4800 Bielefeld 1

(0521) 178260



Inhalt

Der Autor 3

Lizenz 3

Warenzeichen 3

Dateiübersicht..... 3

Der schnelle Einstieg..... 4

 Programm und Benutzer4

 Woraus besteht DipWork?.....4

 Wie konfiguriere ich DipWork?4

 Woraus besteht eine Partie?.....4

 Wie installiere ich eine Partie?5

 Wie werte ich eine Runde aus?.....5

Änderungen zu vorherigen Versionen 6

 zu Version 2.0.6 (22.10.92).....6

 zu Version 2.0.7 (14.2.93).....6

 Installation dieser Version über eine alte6

Programmbeschreibungen 7

DIP_WORK — Arbeitsoberfläche 7

DIP_RUN — Auswerteprogramm..... 7

 Kommandozeilenparameter7

 Auswertungsschritte starten8

 Debug-Parameter8

 Extraprogrammteil Fieldtest.....8

 Extraprogrammteil Orderview8

 Parameter "/mono"8

 Parameter "/"?8

 Funktionen von DIP_RUN.....8

 Der Scanner9

 Der Parser9

 Der Conflicter10

 Der "Winterer" — Rückzüge und Aufbauten10

 Die "Mapper" — Kartenmalen (Malen V6.1).....10

 Die PlainText-Ausgabe.....11

 TeX-Ausgabe11

 Tabulator-Tabelle.....11

 Druckerausgabe.....11

DIP_FLDC — Field-Konverter 11

 Sortiermethode12

 Lesen/Schreiben12

PRINTTIF — Karten ausgeben 12

 Zweck des Programms 12

 Optionen.....13

 Grafikdatei.....13

 Ausgabedatei14

 Treiber14

 Skalierung14

 Parameterdatei.....14

 Grafik-Keyword14

 BGI-Pfad14

 Druck-Resolution14

 Screen-Resolution14

 Increment (Schrittweite der

 Bildschirmausgabe).....14

READTEXT — Textdateien ansehen 15

Ausführliche

Parameterbeschreibung..... 15

Allgemeines..... 15

 Keywords, Parameter15

 Liste (List).....15

 Makro (Macro).....15

 Standardisierung15

 Keyword "END"15

Partieparameter 15

 Regelinterpretationsschalter.....16

 SELF_CUT_OK.....16

 IX_3_INTERPRETATION.....16

 IX_7_INTERPRETATION.....16

 X_INTERPRETATION.....17

 CONVOY_CUTS.....17

 PARTIAL_CUT_POSSIBLE.....17

 Zur Ausgabe in PLAIN.TXT17

 PLAINOUTWIDTH.....17

 TABCOLUMNWIDTH17

 UNDERLINE_ON17

 UNDERLINE_OFF.....18

 COLUMNSEPCODE.....18

 TABLINEENDSTR18

 TABBEGINSTR18

 TABENDSTR18

 Dateinamen und -pfade18

 EDITOR18

FNORDERS.....	18	Einheiten-Bilder.....	29
FNGAME.....	18	Gemalte und leere Karte.....	29
FNVARIANT.....	19	Verwendetes Grafikformat.....	29
VARIANTPATH.....	19		
GAMEPATH.....	19	Beispiel-Varianten zu DipWork.....	29
CURRENTPATH.....	19	1898.....	29
FNFIELDS.....	19	Pseudo-Abstraction II.....	29
FNBORDERS.....	19	Britain.....	30
FNSYNONYMS.....	19	Chaos.....	30
FNPICLIST.....	19	Crowded.....	30
FNMAP.....	19	Pure.....	30
FNERRORS.....	19	Seismic.....	30
FNPLAINOUT.....	19		
FNNEWGAME.....	20	Anfügen einer weiteren Partie an	
FNNEWGAME.....	20	das Projekt	31
FNSAVE.....	20	Installieren von Nicht-Standard-Partien.....	31
FNFIELDTEXT.....	20		
SHELL.....	20	Hinweise für Atari-Benutzer	32
SHELLCOPY.....	20	Systeminterpreter CLI.TOS.....	32
READTEXT.....	20	Texteditoren, Alice.....	32
EVALGAME.....	20	Johannes Schwagereit dazu (27.8.92):.....	32
FIELDENV.....	20	"Can not evaluate, Can not eval, Can not...".....	32
PRINTMAP.....	20	Hinweise von Michael Schröpl, 12.9.92:.....	32
Sonstige Partieparameter.....	21	TTP, PRG und TOS -- Was ist los? [nach	
EXACT_MATCH.....	21	Frage von Torsten].....	32
SPELLCHECK.....	21	DIP_FLDC.....	32
CONVOY_RELEVANT_UNITNR.....	21	Kartenmalen.....	32
NSU_CHECKING_OFF.....	22	PRINTTIF.....	33
ERROR_NR_OUTPUT.....	22	Immer noch Probleme?.....	33
SLOWMOTION.....	22		
VARNAME.....	22	Most Frequently Asked Questions....	33
NAME.....	22	Fragen und Antworten:.....	33
YEAR.....	22		
ZAT.....	22	Syntax für einen scannbaren	
MASTER.....	23	Diplomacy-Befehl	34
SUBMENU.....	23	Schematische Darstellung der	
! am Zeilenende.....	23	Scannerauswertung	35
Beschreibung der Listen	23		
GAMELIST.....	23	Programmlimits.....	35
NATIONLIST.....	23	Maximale Beschränkungen für Werte.....	35
UNITLIST.....	23	Erläuterungen zu Programmfunktionen, die	
ORDERLIST.....	24	keine "Bugs" sind.....	36
MENULIST.....	24	Gelöschte Befehle.....	36
		Landesabkürzungen.....	36
Dateibesreibungen und -beispiele. 25		Presse-Befehl für Einheiten.....	36
Ein Partieverzeichnis	25	IMP Support.....	36
GAME.TXT (oder auch NEWGAME.TXT).....	25	NSO ↔ IMP.....	36
ORDERS.TXT.....	25	Automatischer Abbau.....	36
NEWGAME.TXT.....	25	Befehligen fremder Einheiten.....	36
Aufbau der *.ORD-Dateien.....	26	Verrutschungen in PLAIN.TXT.....	36
Ein Variantenverzeichnis	27		
VARIANT.TXT.....	27	Interesse geweckt?	36
FIELDS.DAT, BORDERS.DAT und			
SYNONYMS.DAT.....	27		
FIELDS.TXT.....	27		
Beschreibung eines Feldes.....	27		
Beschreibung einer Grenze.....	28		
Synonymbeschreibung.....	28		
PictureList.....	28		
Leere Karte (Grafik).....	28		
Grafiken	29		

Dokumentation zum Diplomacy-Projekt

Stand 14.2.93, Version 2.0.7

Der Autor

Ich selber hasse nichts mehr als ellenlange Dokumentationen (wie diese hier), aber ich merke immer mehr, daß man tatsächlich nach dem eingehenden Studium eines solchen Textes um einiges klüger ist als vorher. Wenn ihr also die vollen Möglichkeiten des Programms auskosten wollt, solltet ihr diese Dokumentation studieren. Nicht umsonst habe ich den Punkt "Anfügen einer weiteren Partie" mitten im Text versteckt.

Torsten Will Melanchthonstr. 87 DIE ANTWORT	4800 Bielefeld 1 (0521) 178260 itwill@techfak.uni-bielefeld.de
---	--

Lizenz

Sämtliche Teile dieses Programmpakets sind Public-Domain. Es darf jedoch nur als Ganzes und unverändert weitergegeben werden. Ausnahme: Ergänzungen dürfen gemacht werden, wenn in einer Datei README.TXT auf diese Ergänzungen hingewiesen wird. Das Programmpaket darf nicht zu kommerziellen Zwecken gebraucht werden. In erster Linie ist es zur Unterstützung der Auswertung für Diplomacy-GMs in Postspiel-Zines gedacht. Auch auf Cons (Spieltreffen) kann das Programm Einsatz finden. Jedoch darf das Programm niemals dazu benutzt werden, sich zu bereichern. Also niemals über die Unkosten hinaus Gebühren verlangen!

Schäden an Computer oder Person, die durch das Benutzen dieses Programmpakets auftreten können, sind von jeder Haftung des Autors ausgeschlossen. Die Benutzung des Programmpakets ist nur gestattet, wenn man obigen Bestimmungen zustimmt. Das wird getan, indem man hier unterschreibt:

Warenzeichen

Word, Word für Windows, MS-DOS: Microsoft
IBM PC: IBM
Atari: Atari
WordPerfect: WordPerfect

Turbo Pascal: Borland
Graphics Workshop: Alchemy Mindworks
PaintShow Plus: LogiTech
Diplomacy: Avalon Hill

Dateiübersicht

\Arbeitsverzeichnis	
DIP_WORK.EXE	Projektverwaltungs-Programm
DIP_RUN.EXE	Auswertungsprogramm
DIP_FLDC.EXE	Field-Konverter
READTEXT.EXE	Text lesen
PROJECT.TXT	Projektdatei
\Partie	
GAME.TXT	aktueller Spielstand
ORDERS.TXT	Spielerbefehle
PLAIN.TXT	ausgegebener Auswertungstext
NEWGAME.TXT	neuer Spielstand
*.ORD	Zwischengespeicherte Daten (Orders)
xxxxxxx.TIF	gemalte Karte
\Variante	
VARIANT.TXT	Einstellungen
FIELDS.DAT	Felder-Geografie
BORDERS.DAT	Grenzen-Geografie
SYNONYMS.DAT	alternative Provinznamen
PICLSTxx.TXT	Liste der Einheiten-Bilder fürs Malen
FIELDS.TXT	Quelltext der Felder und Grenzen
xxxxxxx.TIF	leere Karte
\Unitpics	
xx_y_nn.TIF	Ein Bild einer Einheit

Der schnelle Einstieg

Programm und Benutzer

DipWork ist ein Werkzeug, daß das *Auswerten* von **Diplomacy**-Partien mit dem Computer ermöglicht. Die Vorteile sind, daß man (hoffentlich) eine von Fehlern freie Auswertung bekommt, und daß man als Person, die viele Parteien leitet auch Zeitersparnis hat. Das Programmpaket richtet sich weniger an die Spieler von **Diplomacy**, als vielmehr an die *Spielleiter*. Den Spielern kann es nur theoretische Hinweise für vertrackte Situationen in **Dippy** geben.

Das Programm- und Benutzerkonzept ist offen gehalten. Diese beiden Punkte sollen etwas weiter ausgeführt werden:

- Der Benutzer kann sich in das Programm eigene Menüs einbinden, die seinen Anforderungen entsprechen. Durch die immense Zahl an *Parametern* in Dateien, auch für die selbstverständlichsten Dinge, läßt sich Auswertung von *Varianten* **Diplomacy** realisieren. Die Ausgaben des Programms geschehen vorformatiert (~~print~~) und daneben in einer Klartextversion, die alle Daten enthält. Diese Datei (~~newgame.txt~~) kann geeigneten Programm-Tools als Eingabe dienen, um eine den eigenen Ansprüchen gerechte Auswertung zu erstellen.
- Das Programm (~~dp_wakee/tp/pg~~) ist in Module gesplittet, die die einzelnen Phasen einer Auswertung widerspiegeln. Bei der Implementierung von Varianten, die sich nicht durch Ändern von Parametern auswerten lassen, können hier andere Programmteile eingefügt oder aufgerufen werden. Es wurde versucht besonders Computer-abhängige Implementierungen zu vermeiden, so daß eine Portierung theoretisch möglich ist (nur bis Version 2.0.6).

Woraus besteht DipWork?

Die Oberfläche, die sich dem Benutzer präsentiert, wird von ~~dp_wakee/tp/pg~~ bereitgestellt. Von dort aus ruft der Benutzer die verschiedenen Parteien, und für diese wieder die unterschiedlichen Auswertungsschritte auf. Welche das sind, kann persönlich durch die Gestaltung des Auswertungsmenüs modifiziert werden. Die Standardeinstellungen des ganzen Programms, sowie die zu bearbeitenden Parteien, stehen in der Konfigurationsdatei des gesamten *Projekts*, die ~~project~~ heißt.

Die Auswertung selber werden von ~~dp_unee/tp~~ vorgenommen. ~~dp_un~~ besteht aus einzelnen Programmteilen, die wie oben gesagt die einzelnen Auswertungsphasen widerspiegeln. Die möchte ich hier nicht aufzählen, dazu werfe man einen Blick in die Dateibeschreibung (~~dp_wakee/doc~~). Es sei jedoch erwähnt, daß das Programm jeweils eine **Diplomacy-Partie** bearbeitet. Es hat eine gewisse Menge an Eingaben und Ausgaben, und das Dazwischen möchte ich als *Auswertung* bezeichnen.

Da fast alle Ein- und Ausgaben in Dateien festgehalten werden, sollte der Benutzer ein *Textprogramm* zur Verfügung haben. Für den Atari ist ein PD-Programm ~~Ate~~ beigelegt, das diese Aufgabe erfüllt.

Wie konfiguriere ich DipWork?

Wichtig ist, daß man in der Datei ~~project~~ den Pfad dieses Textprogramms hinter dem Schlüsselwort ~~EDITOR~~ aufführt, denn dann kann man diesen von DipWork aus aufrufen, also z.B. "~~EDITOR~~ ~~texteditor~~". Mehr muß man meist nicht tun. Doch sind in ~~project~~ noch einige globale Einstellungen, die man meist nicht verändern muß. Man sollte sich jedoch davon überzeugen, daß verschiedene Parameter die richtigen Programmnamen referenzieren (~~SHELL~~ auf ~~command~~, ~~SHELLCOPY~~ den richtigen ~~copy~~-Befehl ausführt, ~~READTEXT~~ auf ein Textlese-Prg, ~~EVALGAME~~ auf ~~dp_unee/tp~~..., ~~FELDCW~~ auf ~~dp_fcbe/tp~~..., ~~PRINMAP~~ auf ~~print/tp~~...). Für die PC- und Atari-Benutzer sind jeweils in ~~project~~ Beispielschnitte zu finden.

In dem Verzeichnis ~~doc~~ sind einige (viele) Textdateien, die die verschiedenen Funktionen des Programms beschreiben und z.T. von verschiedenen Benutzern als Hilfe für andere Benutzer gedacht ist. Besonders wichtig dürften ~~changes.doc~~ und ~~dp_wakee/doc~~ sein. Ebenfalls nützliche allgemeine Informationen enthalten ~~atari~~, ~~map~~ und ~~readme~~. Für besonders Interessierte hat dieses Verzeichnis (meist) ein Unterverzeichnis ~~other~~, das weiterführende Dokumente enthält, z.B. über andere Varianten, andere Auswertungsprogramme und andere Spielmöglichkeiten.

Atari-Benutzer finden ein Verzeichnis ~~Ate~~, in dem sich ein PD-Texteditor befindet. Einige Hinweise auf seine Benutzung findet man in ~~atari~~, sowie in den bei ~~Ate~~ vorhandenen Dokumentationen. Bevor dieser Texteditor benutzt wird, sind **unbedingt** die Benutzungsbestimmungen von ~~Ate~~ durchzulesen, da dieser nicht zum eigentlichen DipWork-Programm gehört!

Es existiert ein Verzeichnis ~~vars~~ mit mehreren Unterverzeichnissen, wobei jedes die Daten zu einer Variante enthält. Dabei dürfte ~~standard~~ zunächst das Wichtigste sein.

Die anderen Verzeichnisse enthalten Beispielpartien, die als Muster für eigene dienen können. Von besonderer Bedeutung ist dabei vielleicht noch ~~ary~~, denn dort sind die Startdaten einer Standard-Partie abgelegt, die man sich bei Neustart einer Partie kopieren kann.

Woraus besteht eine Partie?

Eine Partie besteht aus Daten, die die *Variante* beschreiben, nach der gespielt wird, aus festen Daten, die während der ganzen Partie gelten und aus Daten die sich von Runde zu Runde ändern. Diesen Erfordernissen werden die hauptsächlichlichen Ein- und Ausgabedateien gerecht (es wird davon ausgegangen, daß die Parameter für Dateinamen nicht geändert wurden):

- Die Variante wird in ~~variant~~, ~~file~~ und ~~board~~ festgehalten. Gehen wir davon aus, daß man eine Standard-Partie auswerten will, dann befinden sich diese Dateien im Verzeichnis ~~vars/standard~~. In ihnen ist unter anderem festgehalten, was für Nationen, Einheitentypen und Befehle es gibt, das Benutzermenü, und vor allem enthalten sie die Geographie der Karte.

Hier für Atari-Benutzer eine Anmerkung: Die Dateien `fetsch` und `bordsch` sind bei der Portierung vom PC problematisch. Es kann sein, daß das Programm absolut jede Auswertung verweigert, weil diese Dateien nicht korrekt sind. Man kann sie für seinen Computer neu erzeugen, wenn man folgenden Programmaufruf ausführt:

`dp_fetsch standardvariant.fetsch`

- Die festen Parameter werden in `game.txt` festgehalten. Dort stehen Partiename, Variante, Mitspieler, der Spielleiter, und evtl. auch der Zugabgabetermin. Diese Datei ist meist in zwei Versionen vorhanden: `game.txt` mit den Daten *vor* der Auswertung, und `newgame.txt` mit denen *nach* der Auswertung. Denn diese beiden Dateien aber auch die
- verändernde Dinge, wie Standorte von Einheiten und Zentrenverteilung. `game.txt` enthält diese *vor* gemachter Auswertung mit DipWork, `newgame.txt` enthält sie *nach* der Auswertung. Weil `game.txt` während der Auswertung nicht verändert wird, kann man diese mehrmals durchführen, weil man z.B. einen Fehler gemacht hat, oder eine Runde wiederholt werden muß. Sobald man die fertige Auswertung akzeptieren will, ist es daher nötig `game.txt` durch `newgame.txt` zu ersetzen! In `newgame.txt` sind noch weitere interessante Dinge festgehalten, die zum Erstellen einer ansprechenden Auswertung genutzt werden können: Sämtliche Spielerzüge, Rückzüge und Aufbauten sind dort noch festgehalten.
- Das Wichtigste für eine Auswertung jedoch, ist die Eingabe der Befehle für die Einheiten! Diese werden durchweg in eine Datei Namens `ordsch.txt` getippt, die das Programm dann liest und die Auswertung durchführt. Desweiteren können hier noch veränderliche Dinge wie aktuelles Jahr und ZAT eingegeben werden.

Die Ausgaben des Programms sind schneller erklärt:

- Zum einen `newgame.txt`, das schon oben erwähnt wurde. Dort ist der neue Stand der Partie festgehalten und außerdem für eine Auswertung verwertbare Information über die Spielerzüge und welche Auswirkungen sie hatten.
- `standardfile` enthält eine grafische Darstellung der Partie, wenn man die Option während des Programmablaufs wählte. Sie kann von vielen Textverarbeitungsprogrammen eingebunden werden. Ansehen oder Ausdrucken kann man sie jedoch auch — weit weniger luxuriös — mit `printfile.bat` oder `typ`.
- Verschiedene `*ord`-Dateien enthalten Zwischenstände der Partie.

Wie installiere ich eine Partie?

Man benötigt ein Verzeichnis für die Partie, in dem die Datei `game.txt` die Anfangsvoraussetzungen der Partie festgehalten sind. Nehmen wir an die Partie soll "Meine" heißen und es sei eine Standard-Partie, dann

- richte man sich das Verzeichnis "`meine`" unter dem Arbeitsverzeichnis von DipWork ein. Dabei ist darauf zu achten, daß der hier vergebene Name nicht länger als 8 Zeichen ist, evtl. Kurznamen ausdenken.
- dann kopiere man aus dem Verzeichnis `and` die Dateien `game.txt` und `ordsch.txt` nach `meine`
- in dem erschaffenen `game.txt` fügt man immer hinter den Schlüsselwörtern `PLAYER` den Namen des Spielers der Nation ein, der das Land führt, z.B. nach der Zeile "`NATIONAL`" für Austria den Spieler "`PLAYER Heinzrad`". Außerdem sollte man noch hinter `NAME` dem Partienamen und hinter `MASTER` den eigenen Namen schreiben.
- in der Datei `project` füge man den Pfadnamen "`meine`" in die Liste der Parteien unter `GAMELIST` ein, Großschreibung ist nicht wichtig, jedoch wieder den kurzen Namen mit maximal 8 Zeichen benutzen!
- fertig

Sicherheitshalber führt man in der Datei `game.txt` noch eine Kontrolle durch:

Alle Einheiten unter "`UNITS`" am richtigen Ort? Haben alle Nationen die richtigen Zentren unter "`CENTERS`"? Sind die Nationen in der richtigen Reihenfolge (alphabetisch)? Sind alle Einheiten und Regionen richtig geschrieben?

Wie werte ich eine Runde aus?

Dazu ruft man das Programm `dp_work` auf. Man befindet sich dann in einem Menü, in dem die Partie "Meine" erscheinen sollte. Wenn nicht, dann hat man in der Datei `project` etwas falsch gemacht. Wählt man den Menüpunkt "Meine" aus, dann kommt man in das Menü speziell für diese Partie. Wie das Menü aussieht, steht in der Datei "`variantstandardvariant`", ist also für alle Parteien der Variante "Standard" gleich. Das erscheinende Bild ist in der nebenstehenden Abbildung angegeben. Kommt man nicht in das Menü, dann existiert wahrscheinlich keine `game.txt`-Datei, oder der darin befindliche Parameter `VARANIPATH` enthält nicht die korrekte Pfadbezeichnung, die in unserem Fall "`VARANIPATH variantstandard`" sein müßte.

Neben dem Menü findet man eigentlich alle Hinweise, die nötig sind, um die Schritte zu einer kompletten Auswertung zu wissen, doch hier noch einmal mit logischer Begründung:

- Wenn dies nicht die erste Runde ist, muß man zunächst die Auswertung der letzten Runde zum akzeptieren, d.h. zur Grundlage der nun folgenden Auswertung machen. Dazu ist es nötig `game.txt` durch das vorhandene `newgame.txt` zu überschreiben. Im Standard-Menü existiert dafür der Menüpunkt "Akzeptieren".

DIP_WORK V2.0.6/PC	by Torsten Will Diplomacy-Standard Rhododendron Funktion wählen	22.10.92
Sinnvolle Abfolge einer Standard-Auswertung: - Auswertung übernehmen - Befehle editieren - Karte kopieren - Scanner, Parser, Con- flicte, Winter, Map, Output für die Partie - Ausgabe nachbearbeiten - Karte drucken - Auswertung an Hrsgbr	>b Edit Befehle k Karte kopieren a Auswertung A Akzeptieren ----- s nur Scanner p nur Parser w nur Winter n nur Mapmaker o nur Output Plain c Output Konflikte K Karte anzeigen D Karte drucken F Felder-Test K Karten-Test v Edit Variante g Edit Standorte u Edit Output S Sicherheitskopie	Tips: Solange NEWGAME.TXT noch nicht über GAME.TXT kopiert ist, kann man die Auswertung wieder- holen. Vor jedem Kartenmalen muß eine leere Karte in das Verzeichnis der Partie kopiert werden.
◀ Editor mit Befehlen aufrufen ▶		

- Nun editiert man die Datei `orderv` mit seinem Lieblings-Texteditor. Dazu kann man den Menüpunkt "Edit Befehle" ausführen (Dazu muß der Editor jedoch in `project` eingetragen sein, siehe oben). Unter die verschiedenen "NATION"-Schlüsselwörter sind die Züge der Einheiten zu schreiben. Wenn ein Spieler einen NMR gemacht hat, gibt man unter der entsprechenden Nation einfach gar nichts ein. Wenn vor dem ersten "NATION" in der Datei die Schlüsselwörter "ZAT", "YEAR" o.ä. auftauchen, dann werden diese Parameter für diese Partie auf die dahinter stehenden Werte neu gesetzt. Die letzte Zeile der Datei sollte "END" sein.
- Möchte man später eine Karte malen lassen, sollte man jetzt noch eine leere Karte (aus dem Variantenverzeichnis) in das Partieverzeichnis kopieren. Das kann man leicht mit dem Befehl "Karte kopieren" bewerkstelligen.
- Nun ist alles fertig für die eigentliche Auswertung: Man ruft sie mit dem bezeichnenden Menüpunkt "Auswerten" auf. Das Programm führt dann alle Schritte nacheinander durch. Evtl. merkt man an den Bildschirmausgaben, daß man einige Züge nicht korrekt eingegeben hat und muß `orderv` noch einmal korrigieren. Man kann die laufende Auswertung jederzeit mit [Ctrl]-[C] oder [Ctrl]-[Break] unterbrechen.
- Ab einem bestimmten Punkt muß man für vertriebene Einheiten die Rückzüge angeben (die schon nach der RHR-Rule vorgeschlagen werden), und wählen ob man Aufbauten zu machen wünscht, weil die Winterkorrekturen anstehen. In dem Fall sind diese ebenfalls noch einzugeben, ebenso wie evtl. Abbauten.
- Zum Schluß der Auswertung steht die Frage ob man eine Karte malen lassen möchte. Beantwortet man diese Frage mit "Ja", muß man vorher, wie oben erwähnt, eine leere Karte kopiert haben. Das Programm gibt dann noch an, wo es Einheiten hinzeichnet, dann ist die Auswertung des Programms beendet.
- In `print` findet man eine vorformatierte Auswertungstabelle, die man in seinen Auswertungstext einbinden kann, ebenso wie eine evtl. gezeichnete Karte. Die Tabelle läßt sich mit Einstellungen in `variant` noch leicht modifizieren (siehe an anderer Stelle in dieser Dokumentation).
- Die Auswertung kann nun ausgedruckt werden und vorher/nachher noch einmal auf Fehler durchgesehen werden.

Änderungen zu vorherigen Versionen

Denjenigen, die mit dem Umgang mit DipWork schon vertraut sind, gebe ich hier eine Kruzbeschreibung der neuen Funktionen dieser Version:

zu Version 2.0.6 (22.10.92)

- 7 PRINTTIF wurde beschleunigt (für Compis ohne CoPro...)
- 7 In PlainOut werden jetzt -SCs nicht mehr mitgezählt
- 7 Bei den Buildups wird der aktuelle Player (Spielernamen) ausgegeben
- 7 Bei Rückzügen/Retreats wird das aktuelle Jahr angezeigt (YEAR)
- 7 Flotte retreat in besetztes Feld mit Subfeldern korrigiert

zu Version 2.0.7 (14.2.93)

- 7 Funktion/Parameter "orderview" in `dip_run` eingefügt
- 7 Zurückgezogene und dann aufgelöste Einheiten werden intern jetzt anders behandelt, dadurch
 - Zurückgezogen und aufgelöste Einheiten in Karte nicht mehr zeichnen
 - Zurückgezogen und aufgelöste Einheiten in PLAIN.TXT nicht mehr angeben
- 7 In PLAIN.TXT werden auch Spielernamen aus mehreren Wörtern ganz ausgegeben
- 7 In PLAIN.TXT auch Ausgabe von Angriffen VOR Rückzügen mit "<" (dadurch 3-Phasenspiel-Ausgabe besser mit /op=conflicted.ord))
- 7 Synonymerkennung für verschiedene Schreibweisen von Feldern eingefügt => neu: Keyword "FNSYNONYMS", Datei "synonyms.dat"
- 7 Ähnlichkeits-Überprüfung bei Feldernamen => neues Keyword "SPELLCHECK"
- 7 Kleiner Fehler bei Aufbauten (Disk Read Error) mancher Varianten
- 7 HomeSC-Nation "All" ermöglicht, in denen jede Nation aufbauen kann
- 7 In UNITLIST in den VARIANT.TXTen sind die AllowedIn= geändert
- 7 Untermenüs in DIP_WORK => Makro 'SUBMENU' in MENULIST ermöglicht
- 7 Eingebautes Makro "'!<CR>" neu => nach Ausführung kein Tastendruck

Bei Installation dieser Version über eine alte

- 7 Vorher PROJECT.TXT und VARIANTS\STANDARD\VARIANT.TXT sichern; nach Installation in den neuen Dateien die selbst gemachten Änderungen neu vornehmen
- 7 Nachher mit Menüpunkt "Service", "Geografie erzeugen" (oder dem Programm DIP_FLDC) FIELDS.DAT, BORDERS.DAT und SYNONYMS.DAT neu erschaffen
- 7 Prüfen ob der neue Partieparameter SPELLCHECK auf das gesetzt ist, was man möchte

Programmbeschreibungen

DIP_WORK — Arbeitsoberfläche

Dieses Programm ist Zuständig für das Verwalten mehrerer Partien. Von hier aus wird am einfachsten eine Auswertung gestartet. Das Programm ruft selbsttätig die notwendigen Programme auf. Im einzelnen macht es folgendes:

- Editieren von PROJECT.TXT
- Partieauswahl
- Menüdarstellung zu einer Partie (siehe MENULIST)
- Ausführen der eines Menüpunktes

Tritt bei der Ausführung einer der Programmteile ein Fehler auf, der den weiteren Ablauf der Auswertung unmöglich machen würde, kehrt das Programm mit einer entsprechenden Meldung vorzeitig zur Partiauswahl zurück.

Dieses Programm greift auf folgende Parameter und Dateien zu:

- Lesen:
- PROJECT.TXT im Hauptverzeichnis
 GAMELIST, FNORDERS, FNGAME, FNVARIANT, EDITOR
- GAME.TXT im Partieverzeichnis
- VARIANTPATH
- ORDERS.TXT im Partieverzeichnis
- VARIANT.TXT im Variantenverzeichnis
 MENULIST
- Partieparameter
 Makros aus Menüs, VARNAME, NAME

DIP_RUN — Auswerteprogramm

DIP_RUN übernimmt so gut wie alle Auswertungsschritte. Es wird sinnvollerweise von DIP_WORK aufgerufen, sollte also im selbstgeschriebenen Menü vorkommen.

Dieses Programm besteht aus mehreren Programmteilen. Insgesamt hat das Programm die Aufgabe, die eigentliche Auswertung durchzuführen. Die einzelnen Programmteile werden im Folgenden getrennt erklärt. Zunächst jedoch eine Beschreibung dessen, wie das Programm aufgerufen wird.

Kommandozeilenparameter

Aufruf: DIP_RUN <gamepath> [<Optionen> ...]

<gamepath> ist der Pfad zu den Partiedaten

Mögliche Optionen sind:

- !<projectfile> Projektdatei; wenn nichts angegeben wird, wird PROJECT.TXT angenommen.
- /s * Scanner starten. Der Scanner ist der Programmteil, der das Testen der Spielerbefehle auf ihre syntaktische Korrektheit übernimmt.
- /p * Parser starten. Der Parser ist der Programmteil, der die Spielerbefehle aufgrund der Situation in der Partie bemängelt und korrigiert, er übernimmt die Prüfung auf das Regeltechnische.
- /c * Conflicter starten. Dieser Programmteil wertet Konflikte in der Partie aus. Hier wird festgelegt, welche Bewegungen, Unterstützungen und Convoys möglich sind, welche nicht und welche Einheiten sich zurückziehen müssen.
- /w * Winterer starten. Dieser Programmteil übernimmt das Auswerten der Rückzüge und Winterkorrekturen, also Auf- und Abbauten, bei der Auswertung.
- /m * Kartenmaler starten. In diesem Programmteil wird eine Grafik erstellt, die als Karte der aktuellen Partie verwendet werden kann.
- /op[=<file>] * PlainText-Ausgabe starten. Startet den Programmteil, in dem die vorformatierte Ausgabe in PLAIN.TXT erstellt wird. Dabei dient <file> als Eingabedatei, und sollte eine *.ORD-Datei sein. Wird hier kein Name angegeben, wird WINTERED.ORD angenommen.
- /all Schaltet alle Optionen um, die mit einem "*" markiert sind.
- /debug1 Gibt zusätzliche Informationen auf dem Bildschirm aus.
- /debug2 Gibt *viel* zusätzliche Informationen auf dem Bildschirm aus.
- /fieldtest Fieldtester starten. Mit diesem Extra-Programmteil kann man Koordinaten, Namen und Grenzen von Feldern testen.
- /orderview Orderview starten. Dieser Extra-Programmteil dient zum Ansehen der *.ORD-Dateien im Klartext.
- /mono Löst Probleme, wenn die Menüs nicht dargestellt werden.
- /? Zeigt die Kurzhilfe.

Alle Optionen sind ein- und ausschaltbar. Standardmäßig ist keine Option aktiv. Durch einmaliges Angeben werden sie eingeschaltet, durch zweimaliges wieder ausgeschaltet usw. Das hat den Effekt, daß man z.B. mit

```
DIP_RUN spiel /all /m
```

Alle Auswertungsschritte für die Partie im Verzeichnis "spiel" durchgeführt werden, außer dem Kartenmaler.

Auswertungsschritte starten

Die Optionen /s /p /c /w /m /op starten die beschriebenen Programmteile, in dieser Reihenfolge. Dies entspricht den Schritten, die nötig sind, um eine Partie vollständig auszuwerten.

Alle diese Optionen auf einmal umschalten kann man, wenn man den Parameter "/all" angibt.

Debug-Parameter

Wenn man auf der Suche nach Fehlern ist ("Warum macht das Programm nicht, was es will"), hilft vielleicht die Ausgabe der (von dem Programmierer oft benutzten) Debug-Information. Diese besteht aus der Ausgabe zusätzlicher Daten, die bei dem normalen Abarbeiten einer Partie noch mehr stören würden, als es das Gewusel auf dem Bildschirm es bisher schon tut.

Sinnvoll ist meines Erachtens jedoch nur "/debug1", dort wird oft dippy-relevante Information ausgegeben. Mit "/debug2" schaltet man den "Dateibearbeitungs-Debug" ein. Man kann dort jedes Öffnen, Schließen, Zuweisen von Dateien, jedes Suchen, Finden, Nicht-Finden nach Schlüsselwörtern verfolgen.

Extraprogrammteil Fieldtest

Das Programm springt nach Eingabe des Parameters "/fieldtest" (statt auszuwerten) in einen speziellen Programmteil, der es erlaubt, die Dateien FIELDS.DAT und BORDERS.BAT auf Herz und Nieren zu untersuchen. Man kann die Namen von Regionen eingeben, und das Programm schmeißt dann alle ihm zu dieser Region bekannten Informationen aus.

Man verläßt diesem Programmteil (und das gesamte Programm) durch Eingabe von "exit".

Extraprogrammteil Orderview

Gibt man den Parameter "/orderview" an, so wird ein Programmteil aufgerufen (statt auszuwerten), mit dem man sich die *.ORD-Dateien genauer und im Klartext ansehen kann. Da diese sonst fast nur Zahlen und fast keinen Text enthalten, ist es manchmal sinnvoll, sich diese Dateien ansehen zu können.

Man kommt dann in ein Menü zu Dateiauswahl, daß man mit ESC verlassen kann. Im oberen Teil des Bildschirms ist dargestellt, was in der gleich folgenden Liste die einzelnen Spalten zu bedeuten haben.

Hat man eine Datei ausgewählt, bekommt man eine Liste der Befehle, die in jener Datei stehen. Was die Spalten bedeuten, wurde a) im Dateiauswahlmenü angezeigt und ist b) dem vorliegenden Dokument der Beschreibung der *.ORD zu entnehmen.

Parameter "/mono"

Standardmäßig ist die Bildschirmausgabe "künstlich" beschleunigt. Sollte man keine Menüs entdecken, wo eigentlich welche auftauchen sollten, dann liegt das daran, daß das Programm nicht erkannt hat, daß Du eine Bildschirmkarte hast, die anders ist als die anderen. Hier hilft die Option "/mono". Manche Hercules-Karten und einige Laptops oder andere ausgefallene Bildschirme könnten solche Schwierigkeiten bereiten — bei der von mir getesteten Hercules-Karte und dem EGA-Laptop klappte jedoch alles wunderbar.

| Diese Option ist auf dem Atari wirkungslos.

Parameter "/"?

Do you speak english? Then you can "get a little help from your friend" with this option.

Funktionen von DIP_RUN

Unabhängig davon, welcher Programmteil aufgerufen wird, greift DIP_RUN auf folgende Daten pauschal zu:

Lesen:

- Partieparameter
keine
- PROJECT.TXT im Hauptverzeichnis
FNORDERS, FNGAME, FN VARIANT, FNORDERS
- GAME.TXT im Partieverzeichnis
- VARIANTPATH
- ORDERS.TXT im Partieverzeichnis
keine
- VARIANT.TXT im Variantenverzeichnis
UNITLIST, NATIONLIST, ORDERLIST
- FIELDS.DAT im Variantenverzeichnis
- BORDERS.DAT im Variantenverzeichnis

Schreiben

- ERRORS.TXT im Partieverzeichnis

Der Scanner

In dieser Programmteil erfolgt die Umsetzung der Züge der Spieler in eine "standardisierte" Form. Es wird nur die Syntax der Befehle überprüft. Etwaige Löschungen werden in ERRORS.TXT ausgegeben. Das Programm liest die Spielerzüge aus der Datei ORDERS.TXT zu dieser Partie. Die standardisierten Befehle werden nach SCANNED.ORD geschrieben.

Seit Version 2.0.7 kann eine Datei SYNONYMS.DAT benutzt werden, in der variierende Schreibweisen für Provinzen angegeben sind. Daher kann man viele Provinzen in unterschiedlichen Schreibweisen angeben.

Ein Befehl eines Spielers muß so aufgebaut sein, wie unter "Syntax für einen scannbaren Diplomacy-Befehl" angegeben ist. Die Abbildungen dort geben im Prinzip nichts anderes wieder, als daß folgende Befehle erlaubt sind (wenn man die Variantendatei der Standardpartie benutzt):

A Bud-Vie	A Boh S A Bud-Vie	En F ENG S Fr A Bre-Lon
Au A Bud-Vie	A Bud S "	F MID-NAT
Armee Bud-Vie	A Boh S Au A Bud-Vie	F MAO-NAO
a bud mve vie	A Sil S A Vie xxx	F PeS-BOT
A Vie xxx	F ENG C A Bre-Lon	F Pet(sc)-GOB
A Vie-xxx	F Nap-Tun	A Afr-Tun
A Vie holds	F TYS C "	A Naf-Tun

England Fleet ENG sup France Armee Bre move Lon

Außerdem hat man mit dem Partieparameter "SPELLCHECK" noch die Möglichkeit, Ähnlichkeiten eingegebener Feldernamen mit einem der vorhandenen Feldernamen zuzulassen. Genauer unter der Beschreibung von "Partieparameter", "SPELLCHECK".

Dieser Programmteil greift auf folgende Parameter und Dateien zu:

Lesen:

- Partieparameter
EXACT_MATCH, FNSYNONYMS, SPELLCHECK
- SYNONYMS.DAT im Variantenverzeichnis
- ORDERS.TXT im Partieverzeichnis
NATION

Schreiben

- SCANNED.ORD im Partieverzeichnis

Der Parser

Hier erfolgt eine Kontrolle der standardisierten Befehle auf ihre Konformität zu den Regeln. Es werden Kontrollen wie doppelte Befehle, fehlende Küsten, fehlende Befehle, falsche Unterstützung, falscher Convoy, unmögliche Convoyroute usw. durchgeführt. Über die getätigten Änderungen wird in am Bildschirm knapp informiert. In ERRORS.TXT stehen Befehle, die evtl. gelöscht worden sind. Buch geführt.

Exakt werden folgende Kontrollschritte durchgeführt (evtl. Änderungen in neueren Versionen):

- Fromfield-Normierung
- Fehlende Befehle (Missing order) (abschaltbar durch "NSU_CHECKING_OFF on")
- Keine solche Einheit (NSU) (abschaltbar durch "NSU_CHECKING_OFF on")
- Subfeld-Normierung und NCS
- Identische Befehle (MUL1)
- Mehrfachbefehle (aber keine Bewegung) (MUL2)
- Mehrfachbefehle (MUL3)
- Unmögliche Convoybefehle
- Bewegungen per Convoy
- Unmögliche normale Bewegung (IMP)
- Unmögliche Bewegung per Convoy (IMP)
- Unmöglicher Unterstützungsbefehl (IMP)
- Kein solcher Bewegungsbefehl beim Convoy (NSO)
- Kein solcher Bewegungs/Haltebefehl bei Unterstützung (NSO)
- Umwandlung von Subfeldern zu Feldern (für Conflicter)
- Löschen des nichtunterstützbar-Flags (für Conflicter)
- Sortieren.

Welche Korrekturen welche Ausgabe in ERROR.TXT hervorrufen, ist in der Beschreibung jener Datei zu entnehmen. Die Korrigierten Züge werden in geeigneter Form in PARSED.ORD geschrieben.

Dieses Programm greift auf folgende Parameter und Dateien zu:

Lesen:

- SCANNED.ORD
- Partieparameter
CONVOY_RELEVANT_UNITNR, NSU_CHECKING_OFF
- GAME.TXT im Partieverzeichnis
NATION, UNITS
- VARIANT.TXT im Variantenverzeichnis
UNITLIST (canConvoy, convoyable)

Schreiben

- PARSED.ORD im Partieverzeichnis

Der Conflicter

Das Auswerten der Konflikte geschieht in diesem Programmteil. Hier wird festgelegt, ob eine Einheit ein Feld betreten darf oder nicht.

Lesen:

- PARSED.ORD im Partieverzeichnis
- Partieparameter
SELF_CUT_OK, IX_3_INTERPRETATION, IX_7_INTERPRETATION, X_INTERPRETATION,
PARTIAL_CUT_POSSIBLE, CONVOY_CUTS, CONVOY_RELEVANT_UNITNR
- VARIANT.TXT
UNITLIST (St=)

Schreiben

- CONFLCED.ORD im Partieverzeichnis

Wenn man im Dreiphasenspiel spielt, dann ist das nach diesem Auswertungsschritt erschaffene "CONFLCED.ORD" als Grundlage für eine PlainText-Ausgabe (siehe dort) zu nehmen.

Der "Winterer" — Rückzüge und Aufbauten

Dieser Programmteil führt Rückzüge durch und erledigt Auf- und Abbauten. Die Rückzüge werden nach der RHR-Rule angegeben, außer bei einem Angriff durch einen Convoy, darauf weist das Programm jedoch hin. Sind Abbauten durchzuführen, ist eine NMR-Reihenfolge nicht festgelegt. Außerdem wird hier die Datei erstellt, die als Grundlage für die nächste Runde dient. Folgendes geschieht im Einzelnen:

- Es schlägt etwaige Rückzüge nach der Justs-Right-Hand-Rule vor
- Es läßt die Wahl zwischen Sommerzug und Winterzug:
- im Winter:
Versorgungsbilanz machen
aufbauen (exklusiv-)oder abbauen
NEWGAME.TXT im Partieverzeichnis erstellen
- im Sommer:
NEWGAME.TXT im Partieverzeichnis erstellen

Lesen:

- Partieparameter
YEAR, ZAT, NAME, MASTER, VARNAME, FNNEWGAME, ERROR_NR_OUTPUT
- GAME.TXT im Partieverzeichnis
NATION, CENTERS
- VARIANT.TXT im Variantenverzeichnis
UNITLIST (AllowedIn=, Costs=)

Schreiben

- NEWGAME.TXT im Partieverzeichnis zur Benutzung als nächstes GAME.TXT
- WINTERED.ORD im Partieverzeichnis

Die "Mapper" — Kartenmalen (Malen V6.1)

Das erfolgreiche (öhöm) Programm MALEN ist in das Auswertungsprogramm eingebunden. Es zeichnet in eine leere Karte an die entsprechenden Koordinaten ein Bildchen einer Einheit. Die so erschaffene Karte kann mit einem geeigneten Programm ausgegeben werden.

- Eine leere Karte muß im Partieverzeichnis vorhanden sein. Diese kopiere man am Besten vorher dort hinein.
- Im Variantenverzeichnis muß eine Datei sein (z.B. PICLSTxx.TXT), in der angegeben ist, wo die Bilder für die Einheiten zu finden sind.
- Für jeden Einheitentyp jeder Nation muß ein Bild existieren.
- Sämtliche Grafiken (Karte und Einheiten-Bilder) müssen im ungepackten TIF-Format vorliegen (und zwar die Version, die den TIF-Header am Anfang der Datei abspeichert, z.B. PaintShow Plus).
- Die erschaffene Karte ist mit einem geeigneten Programm auszudrucken: Word 4.0 oder höher, Word für Windows, Graphics Workshop, Paintshow Plus, WordPerfect 5.1, WordPerfect für Windows, Pagemaker, Coreldraw usw.
- Programme zum Umwandeln der großen TIF-Grafiken (200kB) in gepackte PCX-Bilder (80kB) sind als PD zu haben (zumindest für den PC), wenn Platz auf der Diskette ist, habe ich eines beigelegt.
- Für Atari-Benutzer gibt es bei DipWork zum Drucken dieser Karten das Programm PRINTTIF.

Lesen:

- WINTERED.ORD
- VARIANT.TXT im Variantenverzeichnis
PICLIST, FNMAP
- PicListfile im Variantenverzeichnis
- Einheiten-Bilder aus PicListfile

Schreiben

- Karte im Partieverzeichnis (!)

Die PlainText-Ausgabe

Die Ausgabe der Auswertung in eine Tabelle erfolgt mit diesem Programmteil. Das Ergebnis ist einfach, aber mit einigen Parametern ist die Ausgabe flexibel gestaltbar. Die Ausgabe besteht aus folgenden Teilen:

- Kopfzeile
- Befehle, Tabelle
- Rückzüge, Tabelle
- Aufbauten, Tabelle
- Versorgungszentren, Liste

Die Gestaltung einer Tabelle läßt sich über unten angegebene Partieparameter steuern. Zu einer detaillierten Beschreibung der Parameter schlage man unter dem Punkt "Partieparameter" nach.

TeX-Ausgabe

Die TABxxxSTR-Parameter können prima zur Vereinfachung einer Übernahme der Auswertung in einen TeX-Text benutzt werden. Jedoch ist die Ausgabe noch nicht TeX-reif. Es wäre z.B. sicher sinnvoll, den 'catcode' des Und-Zeichens "&" auf 'normales Zeichen' zu setzen, und als Tabulatorzeichen ein anderes zu benutzen.

Tabulator-Tabelle

Mit "COLUMNSEPCODE 9" kann man den Ausgabertext direkt in ein höheres Textverarbeitungsprogramm übernehmen, daß Tabellen mittels Tabulatoren erstellt. Mit Word für Windows 2.0 wandelt man den Text am Besten in eine Tabelle um (Textstelle Markieren, Menü "Tabelle", Menüpunkt "Text in Tabelle").

Druckerausgabe

Will man eine Ausgabe für den Drucker, ist dieser zweckmäßigerweise zuerst auf 96 Zeichen pro Zeile zu schalten, da sie Ausgabe mit Sicherheit breiter ist als 80 Zeichen. Mit "TABCOLUMNWIDTH 11" und "COLUMNSEPCODE %" bekommt man aber auch eine 80-Zeichen-Ausgabe hin, jedoch werden dann die Fehlerzeichen hinter unterstrichenen Befehlen verschluckt.

Lesen:

- WINTERED.ORD im Partieverzeichnis
- Partieparameter
PLAINOUTWIDTH, TABCOLUMNWIDTH, UNDERLINE_ON, UNDERLINE_OFF,
COLUMNSEPCODE, TABLINEENDSTR, TABBEGINSTR, TABENDSTR
- Schreiben:
- PLAIN.TXT im Partieverzeichnis

Gibt man hier keinen Dateiname als Parameter mit an, so wird als Voreinstellung WINTERED.ORD genommen. Diese Datei enthält normalerweise die Runde nach allen durchgeführten Auswertungsschritten — das heißt für ein normales Zweiphasenspiel! Für ein Dreiphasenspiel wird nach den Konflikten ja eine Auswertung erstellt und die Rückzüge und Auf-/Abbauten als extra Runde gespielt. Das heißt, daß man vor dem Aufrufen des "Winterers" eine Ausgabe haben will. Auch hierzu kann man die PlainText-Ausgabe benutzen. Man gebe als zu benutzende Eingabedatei einfach "CONFLCED.ORD" an, die Datei also, die vor Rückzügen und Auf-/Abbauten vorhanden ist.

DIP_FLDC — Field-Konverter

Im Paket enthalten ist ein Programm, mit dem man sich die Datenfiles der Geographie (FIELDS.DAT, BORDERS.DAT und SYNONYMS.DAT) selber erschaffen kann. Dazu bedarf es einer Eingabe-Textdatei, die so aufgebaut ist, wie FIELDS.TXT, die weiter unten beschrieben ist.

Der Programmablauf des Field-Konverters gliedert sich in vier Phasen:

- Lesen und der Provinzbezeichnungen und -eigenschaften
- Sortieren der Provinzen alphabetisch, damit beim Auswerten (insbesondere des Scanners) schneller ein Feld aufgefunden werden kann, Schreiben von FIELDS.DAT
- Lesen und Zuordnen der Grenzbeschreibungen, Schreiben von BORDERS.DAT
- Lesen und Zuordnen der alternativen Schreibweisen von Provinznamen, Schreiben von SYNONYMS.DAT

Eine Kurzbeschreibung der Funktion erhält man beim Aufruf der Programms mit dem Parameter "/?". Es sei hier erwähnt, wie man das Programm aufrufen muß:

DIP_FLDC <arbeitspfad> <parameterdatei> <quelldatei> [<methode>]

- <arbeitspfad> ist der Pfad in dem sich die <parameterdatei> und die <quelldatei> befinden. In diesen Pfad werden auch die FIELDS.DAT, BORDERS.DAT und SYNONYMS.DAT geschrieben, z.B. "variants\standard".
- <parameterdatei> ist die Datei, in der unten aufgelisteten Schlüsselwörter und Listen gesucht werden, z.B. "variant.txt".
- <quelldatei> ist die Datei, in der der 'Klartext' der Geographie steht, z.B. "fields.txt".
- <methode> ist der zu benutzende Sortieralgorithmus, siehe unten.

Sortiermethode

DIP_FLDC ist durch die Wahlmöglichkeit zwischen schnellen Sortierverfahren ziemlich beschleunigt. Besonders Leute ohne Platten-Cache — oder gar ohne Festplatte — werden sich freuen. Als 4. Kommandozeilenparameter kann man das Sortierverfahren auswählen. Voreingestellt ist "selection"-Sort.

Hier die Sortierverfahren und wann man welches benutzen sollte:

Ich nenne N mal die Anzahl der Felder in FIELDS.TXT. Bei einer Partie mit Subfeldern entspricht eine Vertauschung N Vergleichen. Ein Vergleich benötigt zwei Lese-/Schreiboperationen, eine Vertauschung zusätzlich vier Lese-/Schreiboperationen. Ohne Subfelder fallen die N Vergleiche pro Vertauschung weg!

- "selection" benötigt immer N Vertauschungen und im Mittel $N^2/2$ Vergleiche
- "quick" benötigt $N \log N$ Vertauschungen und im Mittel $2 \ln N$ Vergleiche
- "bubble" benötigt im Mittel $N^2/2$ Tausche und Vergleiche, jedoch viel weniger wenn die Datei schon vorsortiert ist

Man sollte wenn möglich "quick" verwenden. Jedoch ist der ungünstigste Fall für diese Methode eine vorsortierte Datei. Dann dauert es zwar nicht viel länger, doch kann dieser Fall einen Runtime-Error 202 (Stack Overflow) (Atari: 52) verursachen. Das ist nicht weiter schlimm, man sortiere dann mit "Selection"-Sort noch einmal.

Hat man die Datei vorher sortiert, kann man mit "bubble" sortieren. Allerdings bringen Subfelder die interne Reihenfolge durcheinander, so daß bei ihrer Existenz der Vorteil von Bubblesort wieder hinfällig wird.

Im Übrigen richten sich obige Worte vor allem an jene Leute mit langsamen Datenträgern, denn mit zügigem Medium fällt der Unterschied kaum ins Gewicht. Jedoch mußte ich mit Erstaunen feststellen, daß eine RAM-Disk des Ataris eines dieser "langsamen Medien ist", weswegen die Atarianer wenn es geht immer "quick" — oder bei sortierten Feldern ohne Subs "bubble" — verwenden sollten.

Ein Performance-Test, mit zwei Beispielpartien:

Verfahren	Mit Subfeldern		Ohne Subfeldern		Ohne Subs + Vorsortiert	
	Tausch	Read/Write	Tausch	Read/Write	Tausch	Read/Write
Bubble	1831	166477	1579	12168	0	152
Selection	77	13517	71	6136	0	5852
Quick	133	12313	120	1065	76	3458

Lesen/Schreiben

Lesen:

- VARIANT.TXT
UNITLIST, NATIONLIST, FNORDERS, FNBORDERS, FNFIELDS, FNSYNONYMS
- FIELDS.TXT

Feldbeschreibungen (#), Grenzbeschreibungen (–) und Synonymberschreibungen (=)

Schreiben:

- FIELDS.DAT
- BORDERS.DAT
- SYNONYMS.DAT

PRINTTIF — Karten ausgeben

Wichtig: Auf dem Atari erschafft die Standardeinstellung der Ausgabe (nach "PRN") eine gleichnamige Datei "PRN", die noch zum Drucker geschickt werden muß.

Zweck des Programms

Es gibt auf dem PC eine Vielfalt von Programmen, die TIF-Grafiken laden (importieren) und ausdrucken können. Die vom Diplomacy-Programm ausgegebene Karte (z.B. standard.tif) ist im ungepackten TIF-Format abgelegt und kann so ausgegeben werden. Nicht nur das, der Autor empfiehlt ein solches Vorgehen *dringend*.

Das Programm PRINTTIF stellt nur eine Notlösung dar. Für die Atari-Benutzer, die während dem Druck der Grafik ruhig mal 'nen Kaffee trinken gehen können (es reicht wahrscheinlich sogar für eine ganzes Mittagessen) und die PC-Benutzer, die wirklich keine andere Möglichkeit zur Ausgabe haben — und für die Leute, die die Rechenpower in Form eines 386ers aufwärts (mit Festplattencache, wenn möglich) zu Hause stehen haben. Man teilte mir mit, daß auf einem ST das Programm für die Ausgabe einer Standardkarte etwa 2 Stunden benötigen würde, auf einem 286er immerhin etwa eine halbe Stunde. Wer einen 486er hat, bekommt das Ergebnis in guten 5 Minuten.

Die Langsamkeit der Ausgabe liegt hauptsächlich in der Programmiersprache Pascal begründet. Sicher wäre es möglich die Ausgabe zu beschleunigen, doch fehlen dem Autor dazu die nötigen Kenntnisse, außerdem bezweifelt dieser, daß das Programm noch auf den Atari portierbar wäre, und an die Benutzer von jenem Computer richtet sich das Programm ja gerade.

Mit folgenden Programmen ist die Ausgabe einer Dippykarte möglich (ohne Gewähr) (fast alle Programme geschützt):

PRINTTIF, Graphics Workshop, PaintShop, PrintShop, PaintShow Plus, Word für DOS ab 4.0, WordPerfect ab 5.1, Word für Windows ab 1.0, WordPerfect für Windows ab 1.0, Dr. Halo, etc.

Optionen

Mit dem Programm PRINTTIF ist es dem Benutzer möglich, die erschaffenen Diplomacykarten auf einem Ausgabegerät auszugeben. Dies ist besonders für die Atari-Benutzer von Vorteil, da diese nicht über eine andere Möglichkeit verfügen, daß TIF-Format der Grafiken zu benutzen.

Das Programm unterstützt folgende Ausgabegeräte:

screen	Bildschirm (nur PC-Version)
hplj	HP-Laserjet
hpdj500	HP-Deskjet 500
epson9	Epson 9 Nadeln
epson24	Epson 24 Nadeln

Man kann dem Programm auf drei verschiedene Arten seine Voreinstellungen mitteilen: Über Kommandozeilenparameter, über eine Parameterdatei, oder beim Programmablauf. Dabei haben in der Kommandozeile angegebene Optionen Vorrang von solchen, die in der Parameterdatei stehen. Man achte auf Groß- und Kleinschreibung!

Um eine Grafik auszugeben, sind für das Programm folgende Angaben wichtig:

- Name der Grafikdatei
- Gerät/Dateiname der Ausgabe
- Treiber für die Ausgabe
- X- und Y-Skalierung

Weitere Optionen sind bei einigen Ausgabearten wichtig:

- Parameterdateiname
- Keyword für Grafikdateiname in Parameterdatei
- Pfad der Bildschirmtreiber - "BGI-Pfad" (nur bei Treiber "screen")
- Auflösung auf dem Ausgabegerät - "Resolution"
- vertikale Schrittweite - "Increment" (nur bei Treiber "screen")

In der Parameterdatei stehende Optionen sind in der üblichen Form (mit Keywörtern) anzugeben. Eine Besonderheit bildet hier der Kommandozeilenparameter "`=<tif-keywort>`". Mit ihm wird beim Aufruf des Programms bestimmt, nach welchem Keyword in der Parameterdatei gesucht werden soll, um den Namen der Ausgabegrafik festzulegen. Beispiel:

"PRINTTIF !variant.txt =FNMAP"

druckt — wenn man sich im richtigen Verzeichnis befindet - die Grafik aus, die in VARIANT.TXT hinter dem Keyword "FNMAP" angegeben ist, was ja meist die leere Karte dieser Variante ist.

Folgende Tabelle gibt an, wie man welche Parameter über die Kommandozeile und über die Parameterdatei bestimmen kann.

Parameter	Kommandozeile	Beispiel	Parameterdatei
Grafikdatei	#<tif-datei>	#cosa\standard.tif	<i>siehe "<code>=<keyword></code>"</i>
Ausgabedatei	-<ziel>	-printer.prn	PRINTOUTPUT
Treiber	:<treiber>	:hplj	PRINTDRIVER
X-Skalierung	X<zahl>	X1.00	PRINTSCALEX
Y-Skalierung	Y<zahl>	Y1.00	PRINTSCALEY
Parameterdatei	!<text-datei>	!variant.txt	
Grafik-Keyword	=<keyword>	=FNMAP	
BGI-Pfad	\$<pfad>	\$bgi	BGIPATH
Druck-Resolution	R<auflösung>	R300	PRINTRESOLUTION
Screen-Resolution	M+[+]	M++	
Increment	I<zahl>	I1	
Resolutionliste	R?	R?	PRINTRESOLUTION ?
Kurzhilfe	/?	/?	
Weitere Hilfe	/??	/??	

Grafikdatei

Die Grafikdatei muß eine TIF-Datei sein, wie sie in diesem Dokument noch beschrieben wird. Die Ausgabegrafiken von DipWork sind in diesem Format

Ausgabedatei

Seltsamerweise hatte ich zu Hause Schwierigkeiten direkt eine Ausgabe vom Programm an den Drucker anzusteuern. Auf einem anderen Testrechner hatte ich dieses Problem nicht. Abhilfe verschafft eine Ausgabe in eine Datei, z.B. auf die Festplatte, die man dann mittels Kopierbefehl an das Gerät schickt. Wurde unter MS-DOS z.B. die Ausgabe in die Datei "printer.prn" gemacht, so ist die erschaffene Datei folgendermaßen zu Drucker zu senden:

```
COPY /B PRINTER.PRN PRN
```

Treiber

Die zu Anfang erwähnten Treiber sind möglich. Eine Bildschirmausgabe ist auf dem Atari nicht implementiert. Die Treiber für HP-Laserjet und HP-Deskjet 500 sind identisch.

Skalierung

Je nach gewünschter Größe und eingestellter Resolution ist es mitunter notwendig, die Größen des ausgegebenen Bildes den Wünschen entsprechend anzupassen. Dazu kann man mit den Parametern für X- bzw. Y-Skalierung die Grafik strecken oder stauchen. Eine Angabe von "1.00" entspricht einer Ausgabe ohne Größenänderung. Die halbe Breite oder Höhe erreicht man mittels Angabe von "0.5", die doppelte mit "2.0" usw.

Parameterdatei

Damit man nicht immer so viele — immer gleiche — Optionen neu eingeben muß, kann man diese in einer Parameterdatei ablegen. Die Optionen sind, wie in DipWork oft benutzt, mittels Keywörtern dort anzugeben ("END" nicht vergessen). Zu empfehlen ist entweder jeweils eine Datei für die die verschiedenen Ausgabegeräte ("SCREEN.DRV" oder "HPLJ.DRV") oder die Verwendung der VARIANT.TXT.

Optionen, die über die Kommandozeile angegeben sind, haben Vorrang von einer etwaigen Parameterdatei.

Grafik-Keyword

Das hier angegebene Keyword wird in der Parameterdatei gesucht, und der dahinter stehende Dateiname als auszugebene Grafik verwendet.

BGI-Pfad

Bei der Ausgabe mittels des Treibers "screen" ist es notwendig, daß der passende Bildschirmtreiber vorhanden ist. Die Treiber können zusammen in einem Verzeichnis abgelegt sein. Ihre Endung lautet ".bgi". "BGI" ist die Abkürzung für "Borlands™ Graphics Interface".

Eine Grafikausgabe auf dem Atari ist nicht möglich. Folgende Treiber sind im Paket enthalten. CGA-Karten werden nicht unterstützt.

EGAVGA.BGI	EGA-KAarten, VGA-Karten
SVGA16.BGI	VGA 16 Farben 800x600, 1024x768 Punkten (nicht von Borland)
HERC.BGI	Hercules-Karten

Druck-Resolution

Je nach Drucker sind hier eine unterschiedliche Angaben möglich. Anzugeben ist die Dichte in "dpi" - Punkte pro Zoll:

hplj, hpdj500	300, 150, 75
epson9	60, 120, 120d (Doppelte Geschwindigkeit), 240, 40, 90
epson24	60, 120, 90, 180, 360, (8 Nadeln:) 240, 40, 90

Screen-Resolution

Die Auflösung für die Bildschirmausgabe braucht nicht angegeben zu werden — das Programm erkennt Grafikkarte und mögliche Auflösung automatisch. Ausnahme: Einige Super-VGA-Karten unterstützen zusätzliche Grafikmodi, die auch dieses Programm unterstützt. Mit dem Kommandozeilenparameter "M" kann man die Ausgabeauflösung erhöhen.

"M+" 800 mal 600 Punkte (16 Farben, höhö)

"M++" 1024 mal 768 Punkte, non-interlaced/interlaced, je nach Karte und Monitor (auch 16 Farben), 512 KB auf der Karte notwendig.

Increment (Schrittweite der Bildschirmausgabe)

Ich muß leider zugeben, daß die Ausgabe von dem Programm nicht die schnellste ist, daher habe ich für die Bildschirmausgabe ein zusätzliches "Feature" eingebaut. Die anzuzeigende Grafik "schält" sich somit heraus, statt sich langsam von oben nach unten aufzubauen. Man bekommt so schon früh einen Überblick über die Grafik und muß nicht bis zu letzten Zeile warten. Jedoch kann bei ungünstiger Bildlänge die Ausgabe sich selber "überlappen", so daß nur jede zweite oder dritte Zeile dargestellt wird. Die W'keit dafür ist jedoch denkbar gering, zumal ich solche Effekte auch abzufangen versuche. Doch fehlten mir die mathematischen Grundlagen der Ringtheorie, um das auf alle Fälle auszuschließen. Sollte man also mal ein Bild darstellen, das auch nach vollem Bildschirm Aufbau wie eine Jalousie aussieht, benutze man den Parameter "I", um eine Schrittweite von eins o.ä. einzustellen: "I1".

READTEXT — Textdateien ansehen

Zeigt eine Textdatei auf dem Bildschirm an. Der einzige erwartete Parameter ist die anzuzeigende Datei.

Ausführliche Parameterbeschreibung

Allgemeines

Keywords, Parameter

In den Dateien werden Keywords zum Auffinden bestimmter Parameter benutzt. Ein Keyword steht immer am Zeilenanfang (davor dürfen nur Leerzeichen stehen) und ist konventionsmäßig in Großbuchstaben. Durch ein Leerzeichen getrennt folgt dann der Parameter (die Parameter) im geeigneten Format:

Number, Zahl:	Besteht aus einem Zahlenwert, ja nach Zweck in bestimmten Grenzen
Word, Wort:	Eine Zeichenkette ohne Leerzeichen, z.B. Dateinamen, Nationsnamen
Indicator, Anzeiger:	Eine Wort, das einen Zustand/Eigenschaft anzeigt
String, Zeichenkette:	Bis zum Zeilenende gehört alles zu diesem Parameter
Boolean, Ja/Nein:	Ein- oder ausschaltbarer Parameter; mögliche Angaben: "JA, YES, ja, yes, J, Y, j, y, 1, +, on" für eingeschaltet; "NEIN, NO, nein, no, N, , n, 0, -, off" für ausgeschaltet.
ASCII-Liste:	Eine Liste von Zahlen, durch Leerzeichen getrennt. Eine leere Liste wird durch ein Prozentzeichen dargestellt %.

Liste (List)

Ein Sonderfall von Keyword ist das Listen-Keyword (L). Auf ein Listen-Keyword folgen immer die Elemente der Liste, bis zur nächsten Leerzeile.

Makro (Macro)

Ein weiterer Sonderfall ist das Makro. In der "MENULIST" (siehe dort) kann man von Betriebssystem auszuführende Befehle angeben. Um sich Tipparbeit zu sparen, um leichter Änderungen vornehmen zu können oder um partieabhängige Parameter benutzen zu können. kann man dort auf Makros zurückgreifen.

Benutzt wird ein Makro in der Kommandozeile (die 3. Zeile) eines Menüpunktes unter "MENULIST". Man gebe den Namen des Makros zwischen zwei einfachen Hochkommas (z.B. 'MAKRONAME') an und dieser Text wird vor der Ausführung des Kommandos durch die Definition des Makros ersetzt. Statt des Namens eines Makros kann man auch den Namen eines

beliebigen Partieparameters als nehmen, der dann wie eine Makrodefinition interpretiert wird, z.B. 'MASTER' oder auch 'FNGAME'.

Ein Makro wird definiert, indem man selber ein Keyword in eine der Dateien einträgt, wo nach Partieparametern gesucht wird. Der Text hinter dem Keyword wird als die zum Makro gehörige Zeichenkette interpretiert. Einige Makros sind schon vordefiniert (ohne daß sie irgendwo auftauchen), andere sind schon standardmäßig eingetragen; dies ist in der Beschreibung jeweils angegeben.

Standardisierung

Bei vielen Listen wird das erste Wort einer Listenzeile als Standardisierung benutzt; So besteht die Zeile für eine Armee in der UNITLIST z.B. aus den Elementen "A Ar Army Armee", was zur Folge hat, das beim Scannen der Befehle das Auftreten dieser Wörter durch das erste Wort (also "A") ersetzt wird.

Keyword "END"

Am Dateiende von selbstgeschriebenen Dateien sollte man darauf achten, daß man das Wort "END" nicht vergißt. Zwar erkennt das Programm auch so das Dateiende und hat somit ein Äquivalent zu "END", doch wird hinter der letzten Datenzeile ein Zeilenende-Zeichen benötigt. Das Keyword "END" stellt also sicher, daß die letzte Datenzeile mit einem Return beendet wurde.

Partieparameter

Zu jeder Partie gehören einige globale Daten. Einige dieser Daten können wahlweise in einer oder mehreren Dateien angegeben werden. Diese "Partieparameter" werden in folgenden Dateien gesucht (in dieser Reihenfolge):

- ORDERS.TXT im Partieverzeichnis
- GAME.TXT im Partieverzeichnis
- VARIANT.TXT im Variantenverzeichnis
- PROJECT.TXT im Hauptverzeichnis

Sobald der gesuchte Parameter gefunden wurde, wird die Suche nicht mehr fortgesetzt, sondern der gefundene Wert wird verwendet. Möchte man also Partieparameter ändern, die in der Variantendatei definiert sind, braucht deswegen nicht VARIANT.TXT zu ändern. Es reicht, eine entsprechende Zeile in GAME.TXT oder auch ORDERS.TXT einzufügen. Folgende Aufteilung der Partieparameter wäre zu empfehlen:

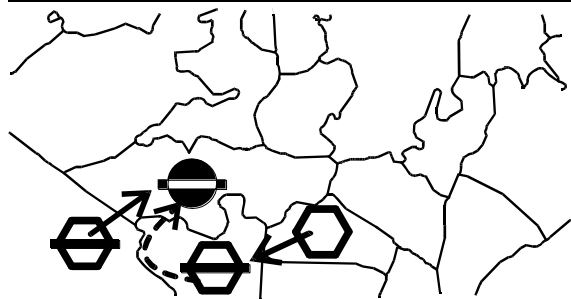
- ORDERS.TXT, Daten, die sich ründlich ändern:
YEAR, ZAT
- GAME.TXT, Daten, die sich während einer Partie nicht ändern:
NAME, MASTER, VARNAME, VARIANTPATH (kein Partieparameter!, muß in GAME.TXT stehen)
- VARIANT.TXT, Daten, die wahrscheinlich für einen GM alle gleich sind
Regelinterpretationsschalter, Dateinamen, EXACT_MATCH, CONVOY_RELEVANT_UNITNR,
NSU_CHECKING_OFF, ERROR_NR_OUTPUT, PlainText-Ausgabe-Optionen, und diverses mehr...

Regelinterpretationsschalter

Einige Regeln lassen sich auf verschiedene Weisen auslegen, andere müssen für einige Varianten anders interpretiert oder geändert werden. Diese variablen Auswertungsregeln können mit diesen Schaltern beeinflusst werden

SELF_CUT_OK

Art	Boolean	Möglichkeiten	ja nein	Standardempfehlung	nein
-----	---------	---------------	-----------	--------------------	------



Ein Angriff einer Einheit auf eine unterstützende Einheit derselben Nation verursacht bei Schalterstellung "ja", daß dieser unterbrochen wird (siehe auch PARTIAL_CUT_POSSIBLE). Die nebenstehende Abbildung veranschaulicht die Züge:

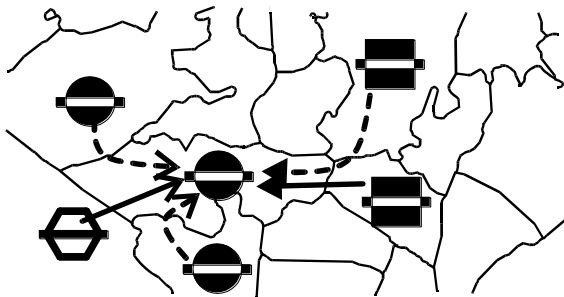
France: F MID-ENG; F Bre S F MID-ENG; A Pic-Bre
England: F ENG xxx

In Schalterstellung "nein" hat der Angriff der A Pic keine Wirkung auf die Unterstützung aus Bre, folglich gelingt der Angriff und En F ENG würde vertrieben werden. Mit "ja" erreicht man jedoch, daß der Support aus Bre abgeschnitten wird, so daß der Angriff aus MID nicht stark genug ist, um

die F ENG zu vertreiben.

IX_3_INTERPRETATION

Art	Zahl	Möglichkeiten	0 1 2	Standardempfehlung	0
-----	------	---------------	-----------	--------------------	---

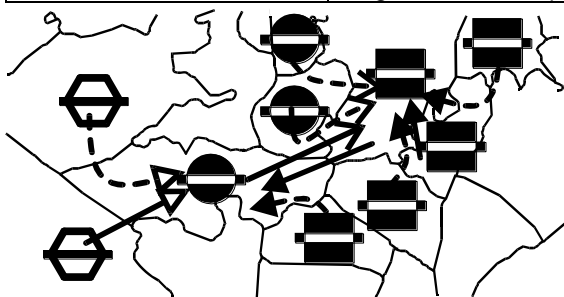


Einheit im Zielfeld stärker als die Verteidigungsstärke des Zielfeldes ist.

Im obenstehenden Beispiel bewegte sich die En F ENG nicht. Die Fr F MID gewinnt den Konflikt in ENG zunächst. Nach Regel IX.3. zählen die Unterstützungen des Engländers zu dem Angriff jedoch nicht bei einer Vertreibung, werden also in einem zweiten Auswertungsschritt ignoriert. Das hat jedoch zur Folge, daß die Ge F Bel den Konflikt in ENG gewinnen würde. Bei Schalterstellung "0" gelingt Fr F MID-ENG, weil "En F ENG trotzdem vertrieben wird". Bei Schalterstellung "1" gelingt weder Fr F MID-ENG noch Ge F Bel-ENG, weil "nicht derselbe Angreifer den Konflikt gewinnt". Bei Schalterstellung "2" gelingt auch keine der Bewegungen: Es wird ohnehin nur Fr F MID-ENG betrachtet und die kann ohne die englischen Unterstützungen nicht vertreiben.

IX_7_INTERPRETATION

Art	Zahl	Möglichkeiten	0 1 2	Standardempfehlung	0
-----	------	---------------	-----------	--------------------	---



Bei gegenseitigen Angriffen zählt ein Angriff nicht für den Konflikt um sein Zielfeld (wird wie ein Haltebefehl ausgewertet), wenn:

- 0 der andere Angriff gelingt
- 1 der andere Angriff stärker ist
- 2 der andere Angriff gleich stark oder stärker ist

In dem nebenstehenden ziemlich unübersichtlichen Beispiel, geht es um die Frage, ob Fr F MID-ENG erfolgreich ist. Dazu wurde folgendes Szenario erstellt: An der Grenze ENG/NTH gibt es einen "Konflikt an der Grenze" (Regel IX.7). Dabei ist En F ENG-NTH 3 stark und Ge F NTH-ENG nur 2 stark. Die englische Einheit gewinnt den Konflikt also. Jedoch

zieht auch Ge F Hol-NTH mit Stärke 3 nach NTH, so daß En F ENG-NTH auch nicht erfolgreich sein kann. Die Fr F MID könnte die En F ENG vertreiben, weil sie Stärke 2 hat, doch ergibt sich im Prinzip dort ein Patt mit der ebenfalls 2 starken Ge F HTH-ENG. Was tun?

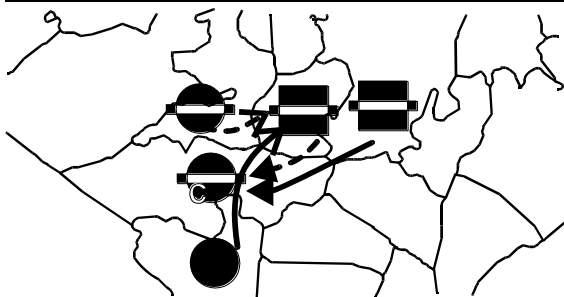
Bei Schalterstellung "0" erreicht die Ge F NTH-ENG ein 2-zu-2-Patt gegen Fr F MID-ENG und die En F ENG wird daher nicht vertrieben, weil "der englische Angriff nicht gelingt". In Schalterstellung "1" jedoch, wird der Befehl der Ge F NTH wie ein Haltebefehl ausgewertet, so daß Fr F MID-ENG erfolgreich ist, denn "der Angriff En F ENG-NTH ist stärker" als der Angriff Ge F NTH-ENG. (Das gleiche geschieht in diesem Fall bei Schalterstellung "3".)

X_INTERPRETATION

Keine Bedeutung mehr. Aus Kompatibilitätsgründen enthalten. Einstellung könnte auf 0 sein.

CONVOY_CUTS

Art	Boolean	Möglichkeiten	ja nein	Standardempfehlung	nein
-----	---------	---------------	-----------	--------------------	------



Wenn in "ja"-Stellung unterbindet auch eine convoyte Einheit eine Unterstützung, andernfalls nicht. In "nein"-Stellung wird das sog. "Pandin-Paradoxon" vermieden. Die neu aufgelegten Dippy-Regeln von Avalon Hill enthalten (meines Wissens) die "nein"-Interpretation.

Betrachtungsweise A: Der Angriff Ge F NTH-ENG ist erfolgreich. Folglich wird En F ENG vertrieben, der Convoy klappt also nicht, und somit auch nicht der Angriff auf Lon. Betrachtungsweise B: Der Angriff En A Bre-Lon vertreibt die Ge A Lon, der Support wird also abgeschnitten. Die F NTH kann En F ENG nicht vertreiben. Dieses Beispiel kann man auch mit zwei sich gegenseitig angreifenden Convoys konstruieren,

wobei völlige Unklarheit über Erfolg oder Mißerfolg von Angriffen entsteht. Die Schalterstellung "nein" verhindert nun, daß En A Bre-Lon den Ge Support abschneidet, so daß Einigkeit darüber zu herrschen hat, daß Ge F NTH-ENG gelingt.

PARTIAL_CUT_POSSIBLE

Art	Zahl	Möglichkeiten	0 1 2	Standardempfehlung	1
-----	------	---------------	-----------	--------------------	---

Wenn nur mit einfach starken Einheiten gespielt wird (normalen Armeen und Flotten), hat dieser Schalter keinen Einfluß.

Beim Spiel mit mehrfach starken Einheiten wirkt sich dieser Schalter aus. Er bestimmt, um welchen Betrag ein Angriff auf eine unterstützende Einheit reduziert wird.

- 0 Jeder gültige Angriff schneidet eine Unterstützung total ab ("cutten")
- 1 Die Stärke des Angreifers wird von der Unterstützung abgezogen
- 2 Jeder gültige Angriff mit Stärke größer null cuttet total

Zur Ausgabe in PLAIN.TXT

In die Datei PLAIN.TXT wird eine vorformatierte Auswertung ausgegeben. Diesen Text möchte man meist noch mit einem geeigneten Textverarbeitungsprogramm nachbearbeiten. Um den diversen gewünschten Formaten gerecht zu werden, kann man die Ausgabe mit einigen Parametern beeinflussen.

PLAINOUTWIDTH

Art	Zahl	Möglichkeiten	etwa 50 bis 255	Standardempfehlung	80
-----	------	---------------	-----------------	--------------------	----

Dieser Parameter gibt die maximale Länge der *Nicht*-Tabellen-Zeilen in PLAIN.TXT an.

TABCOLUMNWIDTH

Art	Zahl	Möglichkeiten	etwa 10 bis etwa 20	Standard-Empfehlung	12
-----	------	---------------	---------------------	---------------------	----

Gibt an, wie viele Zeichen eine Spalte in der Tabelle maximal breit sein kann. Dabei werden die Zeichen die bei "UNDERLINE_ON/OFF" stehen, jedoch mitgezählt. Wenn der Text nicht in die Spaltenbreite hineinpaßt, dann wird er zwischen den Unterstreichungszeichen rechts abgeschnitten. Eine Spaltenbreite von 11 reicht für eine Standardpartie gerade aus, wenn man als jeweils ein einzelnes Unterstreichungszeichen wählt.

Dabei ist noch zu sagen, daß in der Standardpartie die Unterstützungs- und Convoybefehle auf zwei Zeilen umgebrochen werden, so daß auch für sie eine Spaltenbreite von 11 ausreicht.

UNDERLINE_ON

Art	ASCII-Zahlenliste	Möglichkeiten	"%" oder Zahlenliste	Standardempfehlung	91
-----	-------------------	---------------	----------------------	--------------------	----

Diese Codes werden *vor* einem Befehl eingefügt, der nicht durchführbar war. Die Anzahl der Codes zählt mit bei der "TABCOLUMNWIDTH", eine kurze Kette ist daher sinnvoll, wenn man dort einen kleinen Wert eingestellt hat. Eine "91" entspricht einer eckigen Klammer auf: "[". Man kann auch DruckerCodes hier eintragen, z.B. bei einem Epson-Drucker 'unterscheiden an': "27 45 49". Da dies jedoch 3 Zeichen entspricht, sollte man die Spaltenbreite erhöhen. Statt einer Zahl kann man das Prozentzeichen "%" angeben, dann wird nichts eingefügt — das ist aber wahrscheinlich wenig sinnvoll, weil dann nicht erkannt werden kann, welche Befehle ausführbar waren oder nicht.

UNDERLINE_OFF

Art	ASCII-Zahlenliste	Möglichkeiten	"%" oder Zahlenliste	Standardempfehlung	93
-----	-------------------	---------------	----------------------	--------------------	----

Diese Codes werden *nach* einem Befehl eingefügt, der nicht durchführbar war. Die Anzahl der Codes zählt mit bei der "TABCOLUMNWIDTH", eine kurze Kette ist daher sinnvoll, wenn man dort einen kleinen Wert eingestellt hat. Eine "93" entspricht einer eckigen Klammer zu: "]". Man kann auch DruckerCodes hier eintragen, z.B. bei einem Epson-Drucker 'unterscheiden an': "27 45 48". Da dies jedoch 3 Zeichen entspricht, sollte man die Spaltenbreite erhöhen. Statt einer Zahl kann man das Prozentzeichen "%" angeben, dann wird nichts eingefügt — das ist aber wahrscheinlich wenig sinnvoll, weil dann nicht erkannt werden kann, welche Befehle ausführbar waren oder nicht.

[Der "Judge-Diplomacy-Adjudicator" macht hinter nicht ausführbare Befehle ein Sternchen "*", Code 42

COLUMNSEPCODE

Art	ASCII-Zahlenliste	Möglichkeiten	"%" oder Zahlenliste	Standardempfehlung	9
-----	-------------------	---------------	----------------------	--------------------	---

Diese ASCII-Codes werden zwischen zwei Spalten in den Tabelle eingefügt. Damit kann man einen Abstand zwischen den Spalten schaffen, da die "TABCOLUMNWIDTH" ja ganz mit Zeichen aufgefüllt werden kann. Eine "32" fügt ein Leerzeichen ein, "32 32" also zwei, usw. Eine "9" steht für eine Tabulator, recht nützlich für einige Textverarbeitungsprogramme. Letztlich könne die \TeX -Freaks hier z.B. "38" oder "32 38" eintragen, um als Spaltentrennung ein kaufmännisches Und ("&") zu erhalten. Will man keine Zeichen zwischen den Spalten ausgegeben haben, schreibe man statt einer Zahl nur das Prozentzeichen "%" dahinter.

TABLINEENDSTR

Art	Zeichenkette	Möglichkeiten	"%" oder Zeichenkette	Standardempfehlung	%
-----	--------------	---------------	-----------------------	--------------------	---

Hier kann eine Zeichenkette angegeben werden, die *nach jeder Tabellenzeile* ausgegeben werden soll. Ein Wagenrücklauf (CR+LF) wird ohnehin ausgegeben. Aber Programme, die automatisch Tabellen erstellen können, benötigen manchmal noch eine zusätzliche Information am Ende jeder Tabellenzeile. \TeX braucht dort "\cr" oder "\\ " glaube ich. Will man dort nichts extra ausgeben lassen, gebe man nur ein Prozentzeichen "%" an.

TABBEGINSTR

Art	Zeichenkette	Möglichkeiten	"%" oder Zeichenkette	Standardempfehlung	%
-----	--------------	---------------	-----------------------	--------------------	---

In PLAIN.TXT werden insgesamt drei Tabellen erstellt: die Tabelle mit den Zügen, die mit den Rückzügen und die mit den Auf-/Abbauten. Vor diesen Tabellen kann man mit diesem Parameter eine zusätzliche Textzeile ausgeben lassen, die den Beginn der Tabelle kennzeichnet. Damit können sich \TeX -Benutzer den Text ausgeben lassen, den sie vor Tabellen benötigen (ich glaube "\beginab" oder so) oder ähnliches. Die meisten werden dort jedoch wahrscheinlich nichts ausgegeben haben wollen. Dann ist dort nur ein Prozentzeichen "%" hinzuschreiben.

TABENDSTR

Art	Zeichenkette	Möglichkeiten	"%" oder Zeichenkette	Standardempfehlung	%
-----	--------------	---------------	-----------------------	--------------------	---

Wie "TABBEGINSTR" für Ausgabertext *nach* den Tabellen.

Dateinamen und -pfade

Diese Parameter bedürfen selten einer Änderung, doch wer andere Vorlieben für Dateinamen oder -endungen hat, kann diese hier einstellen. vielleicht bedarf auch die Oberfläche, die man benutzt eine andere Endung, um z.B. ein geeignetes Symbol anzuzeigen. Außerdem finden die Dateinamen sicherlich oft Verwendung in der "MENULIST", wenn man sie als Makros benutzt.

Man beachte, daß die Parameter "FNORDERS", "FNGAME", "FN VARIANT" und "EDITOR" keine Partieparameter sind, sondern schon beim Start des Programms DIP_WORK benötigt werden und daher in PROJECT.TXT stehen *müssen*! Außerdem muß "VARIANTPATH" in GAME.TXT stehen.

EDITOR

Art	Wort	Möglichkeiten	Programmname mit Pfad	Standardempfehlung	???
-----	------	---------------	-----------------------	--------------------	-----

Name des Lieblingseditors, mit dem man die diversen zu bearbeitenden Dateien editieren will. Dies ist kein Partieparameter, sondern muß in PROJECT.TXT stehen. Es können keine Batch-Dateien referenziert werden.

FNORDERS

Art	Wort	Möglichkeiten	Dateiname ohne Pfad	Standardempfehlung	orders.txt
-----	------	---------------	---------------------	--------------------	------------

Name der Datei, in dem die Spielerbefehle stehen. Dieser Parameter *muß* in PROJECT.TXT stehen! Die angegebene Datei muß im Verzeichnis der Partie ("GAMEPATH") stehen.

FNGAME

Art	Wort	Möglichkeiten	Dateiname ohne Pfad	Standardempfehlung	game.txt
-----	------	---------------	---------------------	--------------------	----------

Name der Datei, in dem die Standorte und Zentren der Partie stehen. Dieser Parameter *muß* in PROJECT.TXT stehen! Die angegebene Datei muß im Verzeichnis der Partie ("GAMEPATH") stehen.

FN VARIANT

Art	Wort	Möglichkeiten	Dateiname ohne Pfad	Standardempfehlung	variant.txt
-----	------	---------------	---------------------	--------------------	-------------

Name der Datei, in dem Parameter über die gespielte Partie stehen. Dieser Parameter *muß* in PROJECT.TXT stehen! Die angegebene Datei muß im Verzeichnis der Variante ("VARIANTPATH") stehen.

VARIANTPATH

Art	Wort	Möglichkeiten	Pfadangabe	Standardempfehlung	variant\standard
-----	------	---------------	------------	--------------------	------------------

Name der Datei, in dem Parameter über die gespielte Partie stehen. Dieser Parameter *muß* in GAME.TXT stehen! Er muß relativ zum Arbeitsverzeichnis, also dem Verzeichnis in dem DIP_WORK steht, angegeben sein. Man kann ihn natürlich auch als Makro benutzen. Den Gesamtpfad bekommt man dann mit:

'CURRENTPATH\'VARIANTPATH'

GAMEPATH

Art	vordef. Makro	Möglichkeiten	keine	Standardempfehlung	
-----	---------------	---------------	-------	--------------------	--

Dieses vordefinierte Makro steht für den Pfad der auszuwertenden Partie. Es entspricht der Angabe, die man in der Datei PROJECT.TXT in der GAMELIST zu dieser Partie gemacht hat. Die Angabe ist relativ zum momentanen Arbeitsverzeichnis, dem Verzeichnis also, in dem DIP_WORK steht ("CURRENTPATH"). Den Gesamtpfad einer Partie erhält man mit:

'CURRENTPATH\'GAMEPATH'

CURRENTPATH

Art	vordef. Makro	Möglichkeiten	keine	Standardempfehlung	
-----	---------------	---------------	-------	--------------------	--

Dieses vordefinierte Makro steht für das momentane Arbeitsverzeichnis, also das Verzeichnis in dem DIP_WORK steht.

FNFIELDS

Art	Wort	Möglichkeiten	Dateiname ohne Pfad	Standardempfehlung	fields.dat
-----	------	---------------	---------------------	--------------------	------------

Name der Daten-Datei, in der die Informationen über die Felder (Provinzen) der gespielten Karte der Variante abgelegt sind. Dieser Parameter sollte in VARIANT.TXT stehen, damit der Aufruf von DIP_FLDC funktioniert. Die angegebene Datei muß im Verzeichnis der Variante ("VARIANTPATH") stehen.

FN BORDERS

Art	Wort	Möglichkeiten	Dateiname ohne Pfad	Standardempfehlung	borders.dat
-----	------	---------------	---------------------	--------------------	-------------

Name der Daten-Datei, in der die Informationen über die Grenzen der gespielten Karte, bzw. die Zugmöglichkeiten der diversen Einheitentypen der Variante abgelegt sind. Dieser Parameter sollte in VARIANT.TXT stehen, damit der Aufruf von DIP_FLDC funktioniert. Die angegebene Datei muß im Verzeichnis der Variante ("VARIANTPATH") stehen.

FNSYNONYMS

Art	Wort	Möglichkeiten	Dateiname ohne Pfad	Standardempfehlung	synonyms.dat
-----	------	---------------	---------------------	--------------------	--------------

Name der Daten-Datei, in der die Informationen über die alternativen Schreibweisen von Feldnamen abgelegt sind. Dieser Parameter sollte in VARIANT.TXT stehen, damit der Aufruf von DIP_FLDC funktioniert. Die angegebene Datei muß im Verzeichnis der Variante ("VARIANTPATH") stehen.

FNPICLIST

Art	Wort	Möglichkeiten	Dateiname ohne Pfad	Standardempfehlung	piclst64.txt
-----	------	---------------	---------------------	--------------------	--------------

Name der Datei, in dem wiederum die Dateinamen der Bildchen der Einheiten stehen, die zum Kartenmalen benutzt werden sollen. Die angegebene Datei muß im Verzeichnis der Variante ("VARIANTPATH") stehen. Die in der Datei angegebenen Dateinamen sind relativ zum Arbeitsverzeichnis anzugeben, damit mehrere Partien die gleichen Bildchen benutzen können.

FNMAP

Art	Wort	Möglichkeiten	Dateiname ohne Pfad	Standardempfehlung	standard.tif
-----	------	---------------	---------------------	--------------------	--------------

Name der Grafik, die als Karte benutzt werden soll. Beim Kartenmalen erwartet das Programm die angegebene (leere) Karte im Partieverzeichnis. Daher muß sie vor dem Kartenmalen dorthin kopiert werden. Dafür kann man sich z.B. ein Makro schreiben, daß eine leere Karte aus dem Variantenverzeichnis in das Partieverzeichnis kopiert.

FNERRORS

Art	Wort	Möglichkeiten	Dateiname ohne Pfad	Standardempfehlung	errors.txt
-----	------	---------------	---------------------	--------------------	------------

Name der Datei, in die das Programm bei der Auswertung Befehle schreibt, die gelöscht wurden. Dort findet man Befehle, die folgende Fehler verursachten: "Can not evaluate this line", "Multiple orders" und "No Such Unit". Die Datei wird ins Partieverzeichnis geschrieben.

FNPLAINOUT

Art	Wort	Möglichkeiten	Dateiname ohne Pfad	Standardempfehlung	plain.txt
-----	------	---------------	---------------------	--------------------	-----------

Name der Datei, in die das Programm die vorformatierte Auswertung der PlainText-Ausgabe schreibt. Die Datei ist dann im Partieverzeichnis zu finden.

FNNEWGAME

Art	Wort	Möglichkeiten	Dateiname ohne Pfad	Standardempfehlung	newgame.txt
-----	------	---------------	---------------------	--------------------	-------------

Name der Datei, in die das Programm die Standorte der Einheiten und Zentren als Grundlage für die nächste Auswertung schreibt.

FNNEWGAME

Art	Wort	Möglichkeiten	Dateiname ohne Pfad	Standardempfehlung	newgame.txt
-----	------	---------------	---------------------	--------------------	-------------

Name der Datei, in die das Programm die Standorte der Einheiten und Zentren als Grundlage für die nächste Auswertung schreibt.

FNSAVE

Art	Makro	Möglichkeiten	Dateiname ohne Pfad	Standardempfehlung	game.sav
-----	-------	---------------	---------------------	--------------------	----------

Der hier angegebene Dateiname kann für eine Sicherungskopie von GAME.TXT dienen.

FNFIELDTEXT

Art	Makro	Möglichkeiten	Dateiname ohne Pfad	Standardempfehlung	fields.txt
-----	-------	---------------	---------------------	--------------------	------------

Die angegebene Datei dient als "Quelltext" für das Felder-Konvertierungsprogramm DIP_FLDC.

SHELL

Art	Makro	Möglichkeiten	Programmname mit Pfad	Standardempfehlung	c:\dos\command.com
-----	-------	---------------	-----------------------	--------------------	--------------------

Für einige Betriebssystem-Funktionen ist es nötig ein Unterbetriebssystem aufzurufen. Bei der Benutzung von MS-DOS ist das wohl meist "command.com" im Verzeichnis von MS-DOS, beim Atari ist DipWork dafür "CLI.TOS/PRG/TTP" beigelegt. Man kann z.B. normalerweise den "copy"- oder "dir"-Befehl nicht als Programm aufrufen. Zu diesem Zweck muß man "command.com" oder CLI mit dem Parameter "/c" (für command) und dem auszuführendem Kommando angeben. Beispiel: "c:\dos\command.com /c copy eingabe.txt ausgabe.txt". Dafür wird das Shell-Makro also benötigt.

SHELLCOPY

Art	Makro	Möglichkeiten	Zeichenkette	Standardempfehlung	'SHELL' /c copy
-----	-------	---------------	--------------	--------------------	-----------------

Dieses Makro enthält den Befehl, der ausgeführt werden soll, um eine Datei zu kopieren. Dem Makro 'SHELLCOPY' folgen meist noch die Namen der zu kopierenden Quell- und Zieldatei. 'SHELL' ist dabei ein weiteres Makro, siehe dort.

READTEXT

Art	Makro	Möglichkeiten	Programmname mit Pfad	Standardempfehlung	readtext.exe
-----	-------	---------------	-----------------------	--------------------	--------------

Man kann sich Makros schreiben, die das Ansehen von Dateien erlauben. Dabei ist ein Programm aufzurufen, das das gewünschte leistet, z.B. das DipWork beiliegende Programm READTEXT.

EVALGAME

Art	Makro	Möglichkeiten	Programmname mit Pfad	Standardempfehlung	<i>siehe unten</i>
-----	-------	---------------	-----------------------	--------------------	--------------------

Ziemlich wichtig ist dieses Makro, denn es referenziert das zu einer Auswertung auszuführende Programm, in unserem Falle wahrscheinlich meist DIP_WORK.EXE/PRG/TTP/TOS. Man kann den Pfad der auszuwertenden Partie mit dem Makro 'GAMEPATH' schon gleich in die Definition dieses Makros einbinden, Beispiel:

```
EVALGAME DIP_RUN.EXE 'GAMEPATH'
```

Bei der Benutzung des Makros 'EVALGAME' sind dann nur noch die diversen Parameter anzuhängen, die die nötigen Auswertungsschritte angeben.

FIELDENV

Art	Makro	Möglichkeiten	Programmname mit Pfad	Standardempfehlung	dip_fldc.exe
-----	-------	---------------	-----------------------	--------------------	--------------

Dies ist der Programmname des Übersetzers einer Datei wie FIELDS.TXT in FIELDS.DAT und BORDERS.TXT. Man wird hier wohl meist das Programm DIP_FLDC benutzen.

PRINTMAP

Art	Makro	Möglichkeiten	Programmname mit Pfad	Standardempfehlung	<i>siehe unten</i>
-----	-------	---------------	-----------------------	--------------------	--------------------

Das hier referenzierte Programm dient zum Drucken oder Ansehen einer gemalten Karte, wie z.B. das DipWork beigelegte Programm PRINTTIF es kann. Man kann in die Definition dieses Makros schon den Pfad- und Dateinamen der Karte einbinden, dann braucht man bei der Benutzung dieses Makros nur noch die diversen Parameter für Größe, Ausgabegerät, usw. anzugeben. Beispiel:

```
PRINTMAP PRINTTIF.EXE #'GAMEPATH'\FNMAP'
```

Sonstige Partieparameter**EXACT_MATCH**

Art	Boolean	Möglichkeiten	ja nein	Standardempfehlung	nein
-----	---------	---------------	-----------	--------------------	------

Soll in ORDERS.TXT zwischen Groß- und Kleinschreibung unterschieden werden?

In der Schalterstellung "ja" bewirkt der Schalter, daß "A Lon S A Wal xxx" korrekt ist, "a lon s a wal xxx" jedoch nicht korrekt ist. Bei einigen Varianten ist dies sinnvoll, da vielleicht bei den Provinznamen die Großschreibung wichtig ist. So könnte es einen Meeresteil geben, der z.B. "IRE" für 'Irische See' und "Ire" für Irland unterscheidet. Oder die Abkürzungen für die Nationen und Einheiten könnten sich überschneiden, wenn die Groß-/Kleinschreibung nicht unterschieden wird. Dann ist es sinnvoll den Schalter auf "ja" zu stellen.

Man spart jedoch Tipparbeit und macht weniger Eingabefehler, wenn der Schalter auf der Standardeinstellung "nein" steht. Dann werden alle Befehlszeilen aus ORDERS.TXT so interpretiert, daß es egal ist, ob man groß oder klein schreibt.

Der Schalter wirkt sich nur auf ORDERS.TXT aus. Felder und Einheiten in GAME.TXT u.ä. müssen in Groß- und Kleinschreibung bei jeder Schalterstellung korrekt sein.

SPELLCHECK

Art	Zahl	Möglichkeiten	0 oder >= 1	Standardempfehlung	3
-----	------	---------------	-------------	--------------------	---

Noch mehr Freiheiten in der Eingabe in ORDERS.TXT kann man mit diesem Parameter erreichen. Ist sein Wert von größer "0", dann läßt DipWork bei den Feldnamen auch *ähnliche* Schreibweisen zu!

Wurde in der gesamten Tabelle der Felder und Synonyme der exakte Feldname (EXACT_MATCH bleibt wirksam) nicht gefunden, dann werden die beiden Tabellen noch einmal ganz durchsucht, und mit dem bisher nicht gefundenen Feldnamen verglichen. Die dabei gefundene ähnlichste Bezeichnung (wenn es eine gibt), wird als korrekter Feldname verwendet, wenn seine Ähnlichkeit *größer oder gleich* der hinter SPELLCHECK angegebenen Zahl ist.

Das Maß für die Ähnlichkeit, ist die Anzahl der Buchstaben, die der zu suchende Feldname mit einem echten Feldnamen/Synonym am Anfang gemeinsam hat. Beispiele (wenn EXACT_MATCH aus ist):

falsch	richtig	Ähnlichkeit	falsch	richtig	Ähnlichkeit	falsch	richtig	Ähnlichkeit
HEI	HEL	2	Northsea	North	5	WestMed...	West	4
Helgoland	HEL	3	MidAtlant	MID	3	BalticSea	Baltic	6

Wofür soll das gut sein? Nun, in zwei Fällen ist diese Art Auszuwerten sicher recht nützlich:

- Die Karte ist kompliziert und verlangt als Feldbezeichnungen lange Feldnamen. Die meisten Gilgamesch-Karten haben diese Eigenschaft. SPELLCHECK könnte man dann z.B. auf "6" setzen.
- Wenn die Spieler ihre Züge selber eingegeben haben und bei der Auswertung schlecht zu erreichen sind (z.B. per E-Mail), könnten sie ihre Fehlersicherheit erhöhen, wenn sie bevorzugt lange Schreibweisen für Felder benutzen. Übrigens benutzt "Judge, The Diplomacy Adjudicator", ein automatischer Dippy-Auswerter für E-Mail-Partien, ein ähnliches Verfahren.

Gegen eine Ähnlichkeitsprüfung sprechen vielleicht auch zwei Gründe:

- GM- oder Spieler-Tippfehler können falsch korrigiert werden.
- Presse Befehle können verschluckt werden

Die Wahrscheinlichkeit für beides kann reduziert werden, indem man SPELLCHECK einen recht hohen Wert annehmen läßt. Fall b) ist insbesondere recht unwahrscheinlich, da ja nicht der Befehl der Einheit korrigiert wird, sondern nur dort auf Ähnlichkeit geprüft wird, wo ein Feldname zu erwarten ist.

Wichtig: Läßt man die Ähnlichkeit in den Feldnamen zu, so ist bei den speziellen Regeln zu dieser Partie unbedingt darauf hinzuweisen, wenn es in dem *Hausregeln* des Zines nicht schon erklärt ist!

Möchte man die Ähnlichkeitsprüfung nicht haben, dann setze man SPELLCHECK auf "0".

Mit einem hohen Wert für SPELLCHECK einher geht das zur Verfügung stellen einer großen Synonymliste. Wenn die Spieler in einer Variante meist lange Feldbezeichnungen benutzen (müssen), dann sollten in SYNONYMS.DAT mit Wohlbedacht die Feldbezeichnungen gewählt werden. Setzt man SPELLCHECK beispielsweise auf 6, dann sollten viele Synonyme sechs Buchstaben haben, so daß mehrfache Ähnlichkeiten am Wortbeginn verschluckt werden können. So reicht dann z.B. "MidAtl" für sämtliche Bezeichnungen wie "Midatlan", "MidAtlantic", "MidAtlanticOcean", usw. aus.

CONVOY_RELEVANT_UNITNR

Art	Zahl	Möglichkeiten	größer 0 (verschieden)	Standardempfehlung	3
-----	------	---------------	------------------------	--------------------	---

Zu jeder Grenze in BORDERS.DAT (z.B. ENG/NTH) ist zu jedem Einheitentyp angegeben, ob die Bewegung für ihn möglich ist (also im Beispiel "Armee nein", "Flotte ja"), oder was mit diesem Zug passiert. Um beim Convoy Probleme mit Küsten zu lösen, ist ein imaginärer Einheitentyp "Convoyunit" eingeführt worden. Seine Bewegungsmöglichkeiten zeigen an, über welche Grenzen ein einzelner Convoy-Schritt gehen kann (im Beispiel also "ja"). So ist z.B. ein Convoy von LYO nach SpS nicht möglich, wohl aber einer von LYO nach Spa. Außerdem ist beim Spiel mit weiteren Einheitentypen, gewährleistet, daß ein Convoy nur über einheitliche Routen gehen kann. So kann es nicht passieren, daß "Okeaniden" (Amphibienwesen) an Land convoyen.

Die Nummer der Convoyunit ist meist um eins größer, als Einheiten in der UNITLIST angegeben sind. Der Standardwert ist 3, da Armee und Flotte ja wohl wahrscheinlich die Nummern 1 und 2 haben.

NSU_CHECKING_OFF

Art	Boolean	Möglichkeiten	ja nein	Standardempfehlung	nein
-----	---------	---------------	-----------	--------------------	------

Nur für Testphasen geeignet. Ist dieser Schalter auf "ja", dann wird weder geprüft, ob in ORDERS.TXT alle Einheiten angegeben sind, noch ob dort Einheiten stehen, die nicht existieren (in GAME.TXT). Auf Syntax wird weiterhin getestet. So kann man schnell Situationen testen, ohne erst die Standorte der Einheiten eingeben zu müssen. Beim echten Auswerten sollte der Schalter natürlich aus sein.

ERROR_NR_OUTPUT

Art	Boolean	Möglichkeiten	ja nein	Standardempfehlung	nein
-----	---------	---------------	-----------	--------------------	------

In NEWGAME.TXT wird bei Schalterstellung "ja" in geschweiften Klammern die programminterne Fehlernummer ausgegeben, die während der Bearbeitung des Befehls auftrat. Dabei hat jede vom Programm vorgenommene Änderung des Befehls eine Nummer, wie z.B. 100 für einen einfach nicht erfolgreichen Befehl, oder 102 für eine Einheit die sich zurückziehen muß (detaillierte Liste untenstehend). Dabei wird nur die höchste Nummer notiert — wenn ein Fehler mit höherer Nummer auftritt, ersetzt er die alte Nummer (dies ist eine Strategie, die meist klappt; aber eben nicht immer, und daher einer Überarbeitung bedarf).

Fehlernummern von 100 oder größer werden in der Auswertung unterstrichen.

zeNone	= 0;	keine Korrektur aufgetreten	zeTrgMoveOrHoldExp	=130;	Befehl der unterstützten Einheit muß move o. sup sein
zeSupSubFieldChg	= 3;	Zielfeld des unterstützten Befehls geändert (Subfeld)	zeNoOcean	=133;	geleitete Einheit muß im Ozean stehen
zeSubFieldChg	= 6;	Zielfeld des Befehls geändert (wegen Subfeld)	zeUnitCantConvoy	=136;	Einheitentyp (Armee) kann nicht können convoyen
zeSubFromFieldChg	= 9;	Startfeld des unterstützten Befehls geändert	zeIMPMove	=139;	Bewegung unmöglich, keine Grenze (z.B. A Par-Ank)
zeMissingOrder	= 12;	diesen Befehl eingefügt (fehlte also)	zeIMPcmove	=142;	Bewegung per Convoy nicht möglich, keine C-Route
zeUnsuccessful	=100;	nicht erfolgreich	zeIMPSupport	=143;	Bewegung der unterstützten Einheit nicht möglich
zeRetreat	=102;	muß(te) sich zurückziehen	zeNSOSupport	=145;	unterstützter Befehl nicht vorhanden
zeIdenticalRet	=103;	aufgelöst wg. zwei gleichen Rückzugsfeldern	zeNSOconvoy	=148;	geleiteter Befehl nicht vorhanden
zeNoRetReceived	=104;	Keine Rückzugsbefehle erhalten	zeSupMulCoasts	=151;	Zielfeld der unterstützten Bewegung nicht eindeutig
zeOrderExp	=109;	Syntaxfehler beim Befehl (move, hold, ...) der Einheit	zeMulCoasts	=154;	Zielfeld Bewegung nicht eindeutig (z.B. F MID-Spa)
zeMul2Hold	=111;	mehrfache Befehle (keine move) → Haltebefehl	zeMul1del	=200;	identische Befehle → gelöscht
zeMul3Hold	=113;	mehrfache Befehle (mit move) → Haltebefehl	zeMul2del	=203;	mehrfache Befehle (kein move) → gelöscht
zeMoveToField	=115;	Syntaxfehler, Zielfeld der Bewegung ungültig	zeMul3del	=206;	mehrfache Befehle (mit move) → gelöscht
zeSupportedField	=117;	", Startfeld des unterstützten Befehls ungültig	zeNSU	=209;	befehlte Einheit nicht vorhanden
zeSupMoveField	=119;	", Zielfeld des unterstützten Befehls ungültig	zeUnitTypeExp	=300;	Syntaxfehler, Einheitentyp erwartet
zeConvoyedField	=121;	", Startfeld des geleiteten Befehls ungültig	zeUnitOrNationExp	=303;	Syntaxfehler, Einheitentyp oder Nation erwartet
zeConMoveField	=124;	", Zielfeld des geleiteten Befehls ungültig	zeLocationField	=	Syntaxfehler, Standortfeld existiert nicht
zeTrgMoveExp	=127;	Befehl der geleiteten Einheit muß Bewegung sein		306;1	

SLOWMOTION

Art	Zahl	Möglichkeiten	-1 bis etwa 2000	Standardempfehlung	1
-----	------	---------------	------------------	--------------------	---

Seit Version 2.0.2. ist hier die Anzahl der Millisekunden, die nach Ausgabe einer Zeile des Scanners und einem Arbeitsschritt des Parsers gewartet werden soll. Bei zu schnellem Scrolling des Bildschirms bekommt man die Korrekturen, die der Computer macht, gar nicht mehr mit, daher kann man hier eine Verzögerung einstellen. Vielleicht ist "1000" gar nicht so schlecht. 1000 Sollte einer Sekunde entsprechen (laut Compiler), bei einigen schnellen Computern (z.B. meinem 486er) jedoch weniger. Ein Wert von "1" ist die kürzeste Pause, die möglich ist. Ein Wert von "0" unterdrückt zusätzlich noch die Pausen zwischen den Programmteilen, bei denen man eine Taste drücken muß. Ein Wert von "-1" dagegen macht wartet einer Pause von einigen Millisekunden nach der Ausgabe jeder Zeile auf einen Tastendruck.

0 Scrollen ohne Verzögerung und Tastenabfragen, soweit möglich
1-65000 Verzögerung im 1/1000 Sekunden

-1 Wartet immer auf Tastendruck

VARNAME

Art	Zeichenkette	Möglichkeiten	alles Denkbare	Standardempfehlung	Standard
-----	--------------	---------------	----------------	--------------------	----------

Name der gespielten Variante. Diese Angabe hat mit dem Parameter VARIANTPATH (in GAME.TXT) nichts zu tun. So kann man durchaus bei einer Kapitalisten-Dippy-Partie die auf Standard-Partie basiert, hier "Standard-KapDip" hinschreiben.

NAME

Art	Zeichenkette	Möglichkeiten	alles Denkbare	Standardempfehlung	???
-----	--------------	---------------	----------------	--------------------	-----

Hier soll der volle Name der Partie eingetragen werden. Die Angabe hier hat nichts mit dem Parameter GAMEPATH zu tun und kann völlig anders lauten.

YEAR

Art	Zeichenkette	Möglichkeiten	alles Denkbare	Standardempfehlung	siehe unten
-----	--------------	---------------	----------------	--------------------	-------------

Hier kann eine beliebige Zeichenkette eingetragen werden, die der aktuell auszuwertenden Runde entspricht. Das Programm erkennt noch weder selbsttätig wann Frühjahr, Herbst oder Winter ist, noch zählt es die Jahre mit. Hier kann (und soll) also völlig frei eine Bezeichnung für die aktuelle Runde eingetragen werden, wie z.B. "Frühjahr 1901", "Herbst 1902", "F01", "H/W1904" oder ähnliches.

ZAT

Art	Zeichenkette	Möglichkeiten	alles Denkbare	Standardempfehlung	???
-----	--------------	---------------	----------------	--------------------	-----

Hier kann man den Termin eintragen, zu dem die Spieler das nächste Mal ihre Züge abgegeben haben sollen. Was soll man hier noch sagen? Ich trage dort immer "siehe Last Page" ein, weil a) unser Herausgeber den Termin festlegt und b) ich dann nie eine Änderung eintragen muß. Der ZAT erscheint in PLAIN.TXT.

MASTER

Art	Zeichenkette	Möglichkeiten	alles Denkbare	Standardempfehlung	Dein Name
-----	--------------	---------------	----------------	--------------------	-----------

Hier sollst Du Deinem Namen hinterschreiben, der erscheint dann nämlich in PLAIN.TXT

SUBMENU

Art	vord. Makro	Möglichkeiten	Name eines Menüs	Standardempfehlung	???
-----	-------------	---------------	------------------	--------------------	-----

Seit Version 2.0.7 kann man von der MENULIST auch Untermenüs aufrufen, Statt eines auszuführenden Befehls steht in der dritten Zeile des Menüpunktes denn:

```
´SUBMENU´ MENUNAME
```

wobei "MENUNAME" ein beliebiges Schlüsselwort aus VARIANT.TXT ist, dem eine Liste folgt, die genauso aufgebaut ist, wie die Liste unter MENULIST selber. Auch verschachtelte Menüs sind möglich.

! am Zeilenende

Art	vord. Makro	Möglichkeiten	am Zeilenende	Standardempfehlung	
-----	-------------	---------------	---------------	--------------------	--

Wenn in der Kommandozeile eines Menüpunktes (der 3. Zeile) als letztes ein Ausrufezeichen in einfachen Hochkommas steht (also '!'), dann wird nach der Ausführung dieses Menüpunktes nicht auf einen Tastendruck gewartet. Trat jedoch ein Fehler auf, dann wird dennoch gewartet, wie es Standard ist.

Beschreibung der Listen

Auf ein Listen-Keyward folgen immer die Elemente der Liste, bis zur nächsten Leerzeile.

GAMELIST

Diese Liste muß in PROJECT.TXT stehen. Sie enthält die Pfade der installierten Partien für das Menü, das auftaucht, wenn man DipWork startet. Jede Zeile stellt dabei eine installierte Partie dar. Eine Solche Zeile ist wie folgt aufgebaut:

Beispiel
GAMELIST
Cosa % KapDip, die Antwort
Test % Eine Testpartie

Erstes Wort: Pfad zur Partie vom Arbeitsverzeichnis von DipWork aus gesehen. Dieser sollte maximal 8 Zeichen lang sein, wobei Groß- und Kleinschreibung egal ist. Wichtig: Keine Tastenkürzel mit "&" oder "~" definieren!

Optional ein Prozentzeichen "%" mit dahinter folgendem Text: Der Text, der hinter dem Prozentzeichen steht, wird als Kurzinformation in der untersten Bildschirmzeile angegeben, wenn man im Menü auf der Partie steht.

NATIONLIST

Diese Liste befindet sich in VARIANT.TXT.

Das erste Wort jeder Zeile ist die vom Programm später benutzte Abkürzung für die Nation.

Das zweite Wort die Schreibweise des Programms für den vollen Namen.

Weitere Wörter sind alternative Schreibweisen für die Nation in Spielerbefehlen in ORDERS.TXT.

Beispiel:
NATIONLIST
Au Austria Österreich Oesterreich
En England England
Fr France Frankreich
Ge Germany Deutschland
It Italy Italien
Ru Russia Russland Rußland
Tu Turkey Osmanien

Die Reihenfolge der Nationen ist hier wichtig, weil diese mit der in GAME.TXT angegebenen Reihenfolge identisch sein muß. Außerdem ist es die Reihenfolge der Nationen in PLAIN.TXT. Die interne Numerierung der Nationen beginnt mit 1.

UNITLIST

Diese Liste steht in VARIANT.TXT und beschreibt die Eigenschaften der in der Variante vorkommenden Einheitentypen. Ein Einheitstyp wird wie folgt beschrieben:

- 1) mögliche Bezeichnung für diesen Einheitstyp, wobei die erste die vom Programm weiterverwendete ist

- 2) ein Doppelkreuz '#'

- 3) weitere Optionen:

Costs= Versorgungspunkte der Einheit

St= Stärke der Einheit

AllowedIn= Soll eine Einheit in einem Feld aufgebaut werden, dann muß es einen der angegebenen Feldtypen haben.

convoyable Diese Einheit kann convoyen

canConvoy Diese Einheit kann convoyt werden

Beispiel

UNITLIST

```
A Ar Army # Costs=1 St=1 convoyable AllowedIn=AB
F Fl Fleet # Costs=1 St=1 canConvoy AllowedIn=FB
```

"Costs=" ist die Anzahl der Versorgungspunkte, die der Unterhalt dieses Einheitentyps kostet. Normalerweise steht hier "1", aber in anderen Varianten können hier durchaus andere Werte stehen. Man kann die Werte auch in Hundertsteln oder ähnliches angeben, wenn nötig, dann muß man aber darauf achten, die Werte der Zentren in FIELDS.DAT (also in FIELDS.TXT und mit dip_fldc neu erstellen) zu ändern.

"St=" gibst die Stärke der Einheiten dieses Typs an. Normalerweise steht hier eine "1", doch in Varianten kann das anders sein. Auch hier kann man, wenn es halbe Stärkepunkte gibt, einfach multiplizieren, Hauptsache die Verhältnisse der Einheiten zueinander stimmen noch.

Die hinter "AllowedIn=" angegebenen Buchstaben repräsentieren das Gelände, in dem man Einheiten dieses Typs *aufbauen* darf. In FIELDS.DAT (und FIELDS.TXT) hat jede Provinz eine Gruppe an Buchstaben, die ihre Landschaftsarten angibt. Das sind die hier wiederauftauchenden Bezeichnungen. Um eine Einheit in einer Provinz aufzubauen zu dürfen, muß beim Vergleich der Buchstaben der 'Landschaft' und von 'AllowedIn=' *mindestens ein Zeichen* in beiden enthalten sein! (Außerdem ist ein Feld, in dem eine convoyende Einheit stehen muß, ein Feld mit Landschaft "O" (Ocean, Ozean) und das Start- und Zielfeld eines Convoys muß die Landschaft "C" (Coast, Küste) enthalten.)

"canconvoy" ist hinter einer Einheit angegeben, wenn sie die Fähigkeit hat, andere Einheiten zu geleiten. Um geleiten zu können muß die Einheit außerdem in einer Provinz stehen, die als Landschaft "O" (Ocean, Ozean) enthält.

"convoyable" ist hinter einer Einheit angegeben, wenn sie geleitet werden kann. Um geleiten werden zu können, müssen Start- und Zielfeld der Bewegung dieser Einheit den Landschaftstyp "C" (Coast, Küste) enthalten.

ORDERLIST

In dieser Liste, die in VARIANT.TXT steht, sind die unterschiedlichen möglichen Schreibweisen für Einheitenbefehle angegeben. Jedes Wort steht dabei für eine mögliche Schreibweise. Die Anzahl und Reihenfolge der Zeilen ist wichtig, weil an der Zeilennummer erkannt wird, um welchen Befehl es sich handelt.

Zeile 1: Haltebefehl

Zeile 2: Bewegungsbefehl

Zeile 3: Unterstützungsbefehl

Zeile 4: Geleitebefehl

Beispiel

ORDERLIST

```
x xxx hld hold holds hält
- move mve zieht
S support sup supports unterstützt
C convoy con cnv convoys geleitet
```

MENULIST

Diese Liste ist ebenfalls in VARIANT.TXT zu finden.

Das Menü enthält die Zeilen, die bei DipWork zu einer Partie im Menü dargestellt werden. Mit ihm lassen sich die Funktionen steuern, die man auf eine Partie anwenden kann. Dieses Werkzeug ist sehr "mächtig". Wenn man es zu nutzen versteht, kann man sich auf einfachste Weise seine Projekte organisieren. So ist z.B. der Einbau eines KapDip-Programms ebenso möglich, wie das Durchführen von Beben bei Seismic-Dippy.

Ein Menüpunkt ist in je drei Zeilen gefaßt:

1. Zeile: Der Name der Menüpunktes, der als Menüttext erscheinen soll. Schreibt man hier ein Tilde "~" vor einen Buchstaben, so wird dieser Buchstabe als Kurzwahl für den Menüpunkt genommen. Ein Und-Zeichen "&" hat den gleichen Effekt, nur daß das folgende Zeichen nicht angezeigt wird.

2. Zeile: Hier kann man einen Hilfetext (max etwa 70. Zeichen) eingeben, der bei Anwahl des Menüpunktes in der untersten Bildschirmzeile eingeblendet wird.

Beispiel

<diverse Makrodefinitionen>

MENULIST

&Auswertung

gesamte Auswertung durchführen

'EVALGAME' 'GAMEPATH' /all

&iDIR

Zeige Verzeichnis

'SHELL' /c dir |more

~Befehle editieren

Alle Dateien in Texteditor laden

'EDITOR' 'GAMEPATH'\FNORDERS' '!'

&lStandorte Bildschirm

GAME.TXT auf Bildschirm ausgeben

'SHELL' /c type 'GAMEPATH'\FNGAME' |more

Untermeü&u

Das Untermeü aufrufen

'SUBMENU' MEINMENU

3. Zeile: In dieser Zeile ist nun ein DOS-Kommando einzugeben, daß bei Anwahl dieses Menüpunktes ausgeführt werden soll. Man beachte, daß nur exe- und com- Dateien ausgeführt werden können. Residente Befehle (z.B. "dir") und Batch-Dateien müssen mit "command.com /c" aufgerufen werden. Die Endung des aufzurufenden Programms ist explizit anzugeben, ebenso wie der Pfad. Man kann auch Makros (siehe *Partieparameter*, *Makros*) verwenden: Ihre Namen sind in einfache Hochkommas (') zu setzen. Dieser Makroname wird dann durch etwas ersetzt, was irgendwo in den Partiedaten als Parameter hinter einem Schlüsselwort mit dem Namen des Makros auftritt (siehe *Partieparameter*). Zusätzlich gibt es die voreingestellten Makros 'GAMEPATH' und 'CURRENTPATH', die den Pfad zur aktuellen Partie, bzw. den Pfad des Programms darstellen.

Standard ist, daß nach der Ausführung des entsprechenden Kommandos auf einen Tastendruck gewartet wird. Das kann lästig oder nützlich sein. Stellt man als letztes in die Zeile mit dem Kommando ein Ausrufezeichen in Hochkommas ('! '), dann wird nicht auf einen Tastendruck gewartet. Nur wenn ein Fehler bei der Kommandoausführung auftrat, wird trotzdem gewartet.

Statt eines Kommandos kann diese Zeile jedoch auch das vordefinierte Makro 'SUBMENU' enthalten, gefolgt vom Namen eines anderen Schlüsselwortes aus VARIANT.TXT, das ein Untermenü enthält, das ebenso aufgebaut ist, wie die MENULIST. Auch verschachtelte Menüs sind möglich.

Dateibesreibungen und -beispiele

Ein Partieverzeichnis

Jedes Partieverzeichnis enthält Dateien zu einer laufenden Partie. Der Inhalt der Dateien ändert sich meist jede Runde. In dieses Verzeichnis muß man die Datei schreiben, die die Befehle der Spieler enthält (ORDEST.TXT), und man findet bei vollzogener Auswertung hier das Ergebnis des Programmablaufs (NEWGAME.TXT und PLAIN.TXT).

Kommen wir jedoch zur Beschreibung der einzelnen Dateien:

GAME.TXT (oder auch NEWGAME.TXT)

In dieser Datei wird der Stand der Partie zwischen den Runden festgehalten. Es werden neben des globalen Daten zur Partie auch Standorte der Einheiten und Verteilung der Versorgungszentren aufgelistet. Alle Daten in dieser Datei müssen standardisierte Parameter sein (siehe *Parameter*). Ich führe hier nur die für die Eingabe in den Parser notwendigen Daten auf. Benutzt man NEWGAME.TXT als Eingabedatei für irgendein Gestaltungsprogramm einer Auswertung (z.B. die PlainText-Ausgabe von DIP_RUN), kann das entsprechende Programm auch die Daten verlangen, die in dieser Beschreibung jetzt weggelassen werden.

Beispiel	
VARIANTPATH variants\standard	Pfad zur Variante
NATION Au	Beginn der Datenliste für die erste Nation
UNITS	Liste der Einheiten dieser Nation
A Vie	Aufzählung der...
A Bud	...Einheiten...
F Tri	...dieser Nation.
CENTERS	Liste der Zentren dieser Nation
Vie	Aufzählung der...
Bud	...Zentren...
Tri	...dieser Nation.
NATION En	Nächste Nation usw.
UNITS	
A Lpl	
F Edi	
F Lon	
CENTERS	
Vie	
Bud	
Tri	
NATION Fr	
...	
...	
END	Datenende.

In dieser Datei sind alle Nationen aufzuführen, auch wenn sie keine Einheiten oder Zentren mehr besitzen. Desweiteren darf kein 'UNITS' oder 'CENTERS' weggelassen werden, auch ihre Reihenfolge untereinander ist wichtig. Was hinter 'NATION' steht ist völlig egal, da auch die Reihenfolge der Nationen die gleiche sein muß, wie in der Variantendatei unter 'NATIONLIST' angegeben ist.

ORDERS.TXT

Hier muß der GM die Befehle für die Einheiten niederschreiben, die ihm von den Spielern mitgeteilt wurden. Die in dieser Datei geschriebenen Befehle sind in der Scanner-Sprache zu schreiben (siehe *Scanner*).

Beispiel	
YEAR Spring 03	Jahr der Partie (Partieparameter, String)
ZAT normal	Zugabgabetermin (Partieparameter, String)
NATION Austria	Name der ersten Nation (Word)
A Vie xxx	Befehl für erste Einheit
Armee Bud holds	...
f tri	
NATION En	Name der zweiten Nation (Word)
A Lpl-Yor	...
Fleet Lon S A Lpl-Yor	

F Edi greets the pope	...
NATION Fr	
...	
...	
END	Datenende.

NEWGAME.TXT

Diese Datei enthält die vollständige Auswertung einer Runde. Sie kann als Grundlage für einen eigenen Auswertungstext genommen werden. Oder als Eingabedatei für ein Auswertung-Gestaltungsprogramm dienen. Außerdem ist es nötig, vor der nächsten Auswertung diese Datei nach GAME.TXT umzubenennen oder zu kopieren, da sie als Eingabedatei für das nächste Mal verwendet werden sollte. Ich hoffe, daß sich die Zeilen dieser Datei selber erklären. Nach der Lektüre dieser Dokumentation wird das spätestens möglich sein.

Insbesondere: Zu unterstreichende Befehle sind in eckige Klammern gesetzt. Hat ein Land keine Zentren mehr oder keine Einheiten mehr, dann ist die Liste eben leer. Beim Schreiben der Partieparameter (ersten Zeilen der Datei) wird jeweils die gesamte Zeile kopiert. Etwaige Kommentare bleiben also erhalten. Am Ende jeder Liste einer Nation erfolgt eine (genau eine) Leerzeile, die nur ein '%' enthält. Die Datei endet mit 'END'.

Aufbau der *.ORD-Dateien

|Achtung: Wichtige Änderung von bei Version 2.0.6 nach 2.0.7

In den *.ORD-Dateien werden die Partiestände zwischen (und nach allen) den Auswertungsschritten festgehalten. Mit den Informationen in dieser Datei müßte es möglich sein, sich sein eigenes Programm zur Ausgabe einer Auswertung zu schreiben, ohne mühsam auf die Textdatei NEWGAME.TXT zugreifen zu müssen. Die Centers muß man jedoch jener Datei entnehmen.

Die *.ORD-Dateien haben die Spielerbefehle gespeichert. **Die verschiedenen Dateien** enthalten die veränderten Befehle in den unterschiedlichen Phasen der Auswertung:

SCANNED.ORD	umgesetzte Spielerbefehle
PARSED.ORD	korrigierte Befehle
CONFLCED.ORD	Konflikte ausgewertet, Infos über retreats und patts
WINTERED.ORD	Aufgebaut, retreated, Infos über retreats, buildups

Ein Datensatz in dieser Datei besteht aus einem TurboPascal-Record, der wie folgt aufgebaut ist:

TOrder = shortInt;	
TField = record	
field	: word; <i>Nummer des echten Feldes, z.B. von (A) Bul, Mun, (F) SpN, SpS</i>
overfield	: word; <i>Nummer des "Oberfeldes"; identisch mit .field, wenn dieses kein</i>
	<i>Subfeld ist, im obigen Beispiel von (A) Bul, Mun, (F) Spa, Spa</i>
end;	
TOrderLine = record	
Nation	: byte; <i>Nation, der die Einheit gehört</i>
aUnit	: byte; <i>Typ der Einheit</i>
Field	: TField; <i>Nr des Feldes und Oberfeldes des Standortes</i>
Order	: TOrder; <i>Befehl für die Einheit</i>
TrgNation	: byte; <i>Nation der unterstützten/geleiteten Einheit</i>
TrgUnit	: byte; <i>Einheitentyp der " " "</i>
TrgField	: TField; <i>Standortfeld der " " "</i>
TrgOrder	: TOrder; <i>Befehl für die " " "</i>
RefField	: TField; <i>Zielfeld der Bewegung der " " "</i>
Error	: word; <i>Nummer des ("schlimmsten") aufgetretenen Fehlers</i>
Leader	: byte; <i>Nation, die die Einheit befehligte</i>
Comment	: string [80]; <i>Originalbefehl für Auswertung</i>
end;	

Viele **Datenfelder sind nicht immer von Interesse**. Folgende Datenfelder werden nur manchmal benutzt, genaueres siehe unten:

TrgNation	<i>bei Support, Convoy</i>
TrgUnit	<i>bei Support, Convoy</i>
TrgField	<i>immer, außer Halten</i>
TrgOrder	<i>bei Support, Convoy</i>
RefField	<i>bei Bewegungssupport, Convoy</i>

Der **Befehl (Order)** ist als **Zahl abgespeichert**. Dabei bedeuten die möglichen Werte folgendes:

0 empty,	<i>hier steht keine Einheit (nur intern vom Konflikter)</i>
1 hold,	<i>Halten</i>
2 nmove,	<i>Bewegungsbefehl</i>
3 support,	<i>Supports</i>
4 convoy,	<i>Convoys</i>
-1 hsupport,	<i>Haltesupprt (intern Konflikter)</i>
-2 msupport,	<i>Bewegungssupport (intern Konflikter)</i>
-3 cmove,	<i>Bewegungsbefehl, der convoyt wird (CONFLCED, PARSED.ORD)</i>

- 4 umove *nicht erfolgreich bewegt (nmove, cmove)*
- 5 none, *hold, irregular, imp oder missing (intern Konflikter)*
- 6 attacked_from, *Einheit muß sich zurückziehen (nur CONFLCED.ORD)*
- 7 retreat, *Einheit hat sich zurückgezogen*
- 8 pattfield, *Feld duch Patt für ret gesperrt*
- 9 buildup, *Dies ist ein Auf- oder Abbau*

Folgende Dinge werden in den **verschiedenen Fälle in den Datenfeldern** gespeichert:

(Anm: Ich nenne die Zusammenfassung von "Nation, Einheitstyp, Feld" immer "Location" einer Einheit.)

Haltebefehl: Location der befehligten Einheit Order = hold Comment = Originalbefehl für Auswertung Bewegungsbefehl: Location der befehligten Einheit Oder = nmove TrgField = Zielfeld der Bewegung Comment = Originalbefehl für Auswertung Support zur Bewegung: Location der befehligten Einheit Order = support (oder msupport in PARSED.ORD) TrgXXXX = Location der unterstützten Einheit TrgOrder = nmove RefField = Zielfeld der Bewegung der unterstützten Einheit Comment = Originalbefehl für Auswertung Support zum Halten: Location der befehligten Einheit Order = support (oder msupport in PARSED.ORD) TrgXXXX = Location der unterstützten Einheit TrgOrder = hold Comment = Originalbefehl für Auswertung Convoy: Location der befehligten Einheit Order = convoy TrgXXXX = Location der geleiteten Einheit TrgOrder = nmove (oder cmove) RefField = Zielfeld der Bewegung der geleiteten Einheit Comment = Originalbefehl für Auswertung	Rückzug in ein Feld: Location der befehligten Einheit Order = retreat TrgField = Zielfeld des Rückzugs (s. "Abbau nach Rückzug"!) Comment = Originalbefehl für Auswertung Abbau nach Rückzug: (Sonderfall; geändert in V2.0.7) Location der befehligten Einheit Order = retreat TrgField = 0, 0 TrgOrder = buildup RefField = Zielfeld des Rückzugs und danach Feld des Abbaus Comment = Originalbefehl für Auswertung Rückzug-ex: Location der befehligten Einheit Order = retreat TrgField = 0, 0 Comment = Originalbefehl für Auswertung Aufbau: (zusätzlich zum normalen Befehl) Nation = Aufbau von Nation... Order = buildup aUnit = Aufgebauter Einheitentyp Field = 0, 0 TrgField = Feld des Aufbaus Comment = nicht definiert Abbau: Location der abgebauten Einheit Order = buildup (siehe aber auch "Abbau nach Rückzug"!) TrgField = 0, 0 Comment = Originalbefehl für Auswertung
---	--

Ein Variantenverzeichnis

VARIANT.TXT

Die Datei dessen Namens einhält globale Daten zu einer Partie. Die Daten dieses Typs ändern sich meistens nicht im Laufe einer Partie. Da viele Parteien mit den Parametern laufen, sind sie in dieser Datei zusammengefaßt. Auf diese Datei greifen dann alle Parteien zu, die die entsprechende Variante hinter dem Keyword "VARIANTPATH" angegeben haben. Eine ebenfalls ausführliche Dokumentation der verwendeten Parameter und Listen findet man in dieser Dokumentation unter Der Beschreibung der Partieparameter und Listen.

Die Datei enthält folgende Dinge:

NATIONLIST, UNITLIST, ORDERLIST, MENULIST

Regelinterpretationsschalter, diverse Dateinamen, sonstige Parameter

Die Datei endet mit einem "END" vor dem END sollte eine Leerzeile stehen, weil davor vielleicht eine Liste steht.

FIELDS.DAT, BORDERS.DAT und SYNONYMS.DAT

Diese Dateien enthalten die Beschreibung der Geographie der Partie, bzw. der benutzten Karte. Die Daten darin sind binär abgespeichert und werden mit dem Field-Konverter DIP_FLDC aus Textdateien wie FIELDS.TXT erschaffen. Man kann den Inhalt dieser beiden Dateien testen und auf Korrektheit prüfen, indem man den Extra-Programmteil des Fieldtests von DIP_RUN aufruft (siehe dort).

FIELDS.TXT

In dieser Datei ist die Geographie der Karte abgelegt. Es werden sowohl die Felder, wie auch die Grenzen beschrieben. (Auch diese Datei sollte mit einem "END" aufhören. Die Datei muß, bevor die Geographie benutzt werden kann, noch in die Binären Dateien FIELDS.DAT und BORDERS.DAT übersetzt werden. Dies geschieht mittels des Programms DIP_FLDC (siehe dort).

Beschreibung eines Feldes

Bud 860 810 - L 1 Au

- 1) Keyword "#" für Feldbeschreibung
- 2) Feldname;

darf folgende Zeichen enthalten: Kleinbuchstabe, Großbuchstabe, Ziffer, _ / () : , = ^

keine Leerzeichen und Bindestriche!

3), 4) x- und y-Koordinate auf der Karte

5) Ist eine Küste von... ("- für: ist keine Küste)

6) Feldtyp; Prinzipiell beliebig, hängt von Anwendung ab.

Die Buchstaben werden ohne Leerraum aneinander geschrieben. Einige Buchstaben haben schon eine feste Funktion:

"O"cean. Möglicher Standort einer convoyenden Einheit

"C"oast. Möglicher Convoystart und Convoyziel.

"L"and. Im Moment keine programmtechnische Bedeutung.

Desweiteren werden hier die Landschaftstypen angegeben, die ein Aufbauen von Einheiten erlauben (siehe "UNITLIST",

"AllowedIn="). In der Standardvariante werden dafür folgende Buchstaben benutzt:

"A"rmy. Aufbau einer Armee möglich

"F"leet. Aufbau einer Flotte möglich

"B"oth. Beides zum Aufbau möglich

7) Versorgungspunkte (siehe "UNITLIST", "Costs=")

8) Heimatversorgungszentrum der Nation XY,

"All" heißt hier, daß jede Nation hier aufbauen kann, wenn sie das Zentrum besitzt

Beschreibung einer Grenze

Beispiel					
-	Kie	Mun	ja	nein	Nur Armee darf das
-	Kie	Hol	ja	ja	Armee und Flotte dürfen das
-	Kie	HEL	nein	ja	Nur Flotte darf das
-	Mun	Bur	ja	-	Startfeld unbetretbar für Flotte
-	NTH	Lon	-	ja	Startfeld unbetretbar für Armee
-	Por	SpS	Spa	ja	für Armee Subfield-change, Flotte ja
-	Gas	Spa	ja	SpN	Armee ja, für Flotte Subfield-change
-	Por	Spa	ja	imp	Armee ja, Flotte mehrere Küsten
-	Spa	Mar	ja	-	Armee ja, für Flotte existiert Startfeld nicht
-	SpS	Mar	-	ja	für Armee existiert Startfeld nicht, Flotte ja

1) Keyword "-" für Grenzenbeschreibung

2) Name des Startfeldes einer Bewegung

3) Name des Zielfeldes einer Bewegung

4) Bewegung möglich für Einheitstyp eins (siehe UNITLIST)?

5) Bewegung möglich für Einheitstyp zwei?

...

dabei ist eine der folgenden Angaben für eine *Bewegung* (bei 4), 5), usw.) möglich:

"ja" Die Bewegung ist möglich

"nein" Die Bewegung ist nicht möglich

"-" Startfeld existiert für Einheitstyp nicht (also Fehler)

"imp" Bewegung nicht möglich, da mehrere Küsten möglich wären

Feldname Bewegung möglich, jedoch Änderung des Zielfeld-Namens

Tip:

Es empfiehlt sich, in der Standardpartie als dritten Einheitstyp eine "convoyende Einheit" einzufügen, da die Schritte, die während eines Convoys aufgrund von Küsten nicht identisch sind, mit den Bewegungen einer Flotte. CONVOY_RELEVANT_UNITNR muß dann 3 sein.

Synonymbeschreibung

Beispiel		
=	MID MAO	"MAO" ist eine alternative Schreibweise von "MID"
=	Pet StP	"StP" steht für "Sankt Petersburg"
=	PeN Pet(nc) Pet/nc	Petersburg-Nordküste hat mehrere alternative Schreibweisen
=	PeN StP(nc) StP/nc	...die man auch auf mehrere Zeilen verteilen kann

1) Keyword "=" für Synonymbeschreibung

2) Name des Feldes, dessen Synonyme in der Zeile angegeben werden

3), 4), 5),... alternative Schreibweisen des Feldnamamens

PictureList

A Fr VARIANTS\UNITPICS\fr a 64.tif

In einer gesonderten Datei sind die Namen der Grafiken abgelegt, die die Bilder der Einheiten enthalten. Damit mehrere Varianten die gleichen Bilder benutzen können, sind diese in einem meist in einem speziellen Verzeichnis abgelegt. Im Variantenverzeichnis existiert nur eine Liste der Bilder mit den Pfaden und Namen dieser Grafiken. Von den Einheiten-Bildern sind DipWork zwei verschiedene Größen beigefügt: 40-Pixel und 64-Pixel Breite. Für beide Bildserien existiert jeweils eine PictureList-Datei mit den entsprechenden Dateireferenzen

Die Wörter jeder Zeile in der PictureList-Datei bedeuten

1) Einheitstyp

2) Nation

3) Voller Dateiname mit Pfad (vom Programm aus) zu dem

Leere Karte (Grafik)

Nicht notwendigerweise, aber ebenfalls im Variantenverzeichnis befindet sich die TIF-Grafik, die die leere Karte dieser Variante darstellt. Diese wird hier nur benötigt, um sie vor jedem Kartenmalen in das gewünschte Partieverzeichnis zu kopieren.

Grafiken

Einheiten-Bilder



In einem eigenen Verzeichnis befinden sich die Bilder der Einheiten, die beim Malen einer Karte an die entsprechenden Stellen der leeren Karte kopiert werden. Das normale Verzeichnis heißt "VARIANT\UNITPICS". Die hierin stehenden Bilder werden von den in der PictureList (siehe dort) stehenden Dateinamen referenziert.

Außer den Standard-Nationsbildern in zwei Größen (AU_A_64.TIF bis TU_64.TIF und AU_A_40.TIF bis TU_40.TIF) gibt es noch welche für zwei weitere Nationen, die man bei Bedarf verwenden kann. Diese liegen jedoch nur in der Größe von 64 Pixeln vor (X1_A_64.TIF, X1_F_64.TIF, X2_A_64.TIF, X2_F_64.TIF).

Gemalte und leere Karte

Es wurde schon mehrfach in dieser Dokumentation erwähnt, aber es sein noch einmal gesagt: Um die Funktion des Kartenmalens durchzuführen, muß eine leere Karte (d.h. eine Karte, die nur die Grenzen u.ä. enthält, und noch keine gemalten Einheiten). in das entsprechende Partieverzeichnis kopiert werden. Während des Kartenmalens wird diese Grafik dann verändert. Danach kann die kopierte und veränderte Datei irgendwie ausgegeben werden, z.B. mit einem geeigneten Mal- oder Textverarbeitungsprogramm, oder mit PRINTTIF.

Eine leere Karte wird mit DipWork in den entsprechenden Variantenverzeichnissen geliefert. Dort steht sie nur zu dem Zweck, wie eben beschrieben irgendwann kopiert zu werden. Sonst verändert kein Programmteil von DipWork die im Variantenverzeichnis stehende Karte.

Verwendetes Grafikformat

Alle vom Programm verwendeten Grafiken sind im umgepackten TIF-Format abgespeichert. Es wird die TIF-Version benötigt, die den Header am Anfang der Datei ablegt (Byte \$04 = \$08). Dieses verwendete Format wird von einer Vielzahl von Programmen unterstützt -- zumindest zu Lesen. Leider schreiben viele Programme, auch wenn das Dateiformat ungepackt ist, den Header nicht an den Anfang. Das einzige Programm das ich kenne, daß dies beides machen kann, ist PaintShow Plus.

Beispiel-Varianten zu DipWork

Dem Paket DipWork liegen einige Beispielvearianten bei. Deren Regeln und Startbedingungen sind im Folgenden genauer erklärt. Außerdem noch Hinweise bei der Arbeit mit DipWork bei den spezifischen Varianten.

Wenn <Variante> der Name einer Beispiel-Variante ist, dann findet man Verzeichnis "VARIANTS\<Variante>" immer die folgenden Dateien:

- README.TXT Erläuterungen und Regeln zu dieser Variante
- FIELDS.TXT die man mit dem Menüpunkt "Service", "Geographie erstellen" in FIELDS.DAT, BORDERS.DAT und SYNONYMS.DAT übersetzen kann (oder mit "DIP_FLDC variants\<Variante> variant.txt field.txt")
- GAME.TXT Startpositionen der Einheiten
- ORDERS.TXT erste Beispiellrunde

Manchmal kann man die Standardkarte zum Kartenmalen verwenden. Darauf wird besonders hingewiesen. Die Datei STANDARD.TIF liegt jedoch nicht in jedem Variantenverzeichnis vor, sondern man kann sich variants\standard\standard.tif in das gewünschte Variantenverzeichnis kopieren.

Die Varianten *1898*, *Britain*, *Chaos*, *Crowded* und *Pure* sind den Regeln des "Diplomacy Adjudicator" entnommen.

1898

In der 1898-Variante startet das Spiel nach den Winteraufbauten von 1898, in denen jede Nation eine Einheit hat. Jede Nation muß seine anderen Zentren erst erobern, bevor es darin aufbauen darf. Das Die Siegbedingung bleibt 18 Versorgungszentren.

Die Start-Standorte der Einheiten sind:

Austria A Tri	France A Bre	Italy A Nap	Turkey A Smy
England F Edi	German A Kie	Russia A Pet	

Die Grafik der Standardkarte kann verwendet werden.

Pseudo-Abstraction II

Die hier enthaltenen Daten ABSTRA2.TIF und FIELDS.TXT sind dazu gedacht, eine Standard-Partie auf einer Abstraction II - Karte zu spielen. "Hä, Warum das?" Man spiele auf der Abs II - Karte eine Partie nach den Standard-Diplomacy-Regeln, wenn man Lust hat. Man beachte, daß es in den Abs II - Regeln keine Küsten gibt. Folglich sind diese auch in der Geographie für das Programm nicht berücksichtigt.

Sollte man sich eine Karte malen lassen wollen, empfehle ich, die nur 40 Punkte großen Einheiten (statt der 64 Punkte großen), die passen besser in die Karte. Man verwende die Karte VARIANTS\ABSTRA2\ABSTRA2.TIF.

Die Geografiedaten sind allerdings sozusagen "Version 1.0 beta", weil die Korrektheit der Grenzen nur von zwei menschlichen Augen kontrolliert wurde. Wer Zeit und Lust hat, kann eine weitere Kontrolle durchführen, ich wäre dankbar. Ab besten frage man mich aber vorher noch mal, vielleicht gibt es inzwischen ja schon eine mehrfach kontrollierte Version.

Dabei möchte ich auch gleich anmerken, daß ich sehr (sehr, sehr) gerne Vorschläge haben möchte, welche weiteren Dippy-Varianten man in das Projekt einbinden sollte, die möglichst (aber nicht unbedingt) mit dem Standard-Befehlssatz (hold, move, S, C) auskommen. Ich habe die Möglichkeit die entsprechenden Grafiken zu erstellen. Die Geographie könnte jemand anders übernehmen, so kann man mit der Zeit viele Varianten einbinden.

Britain

In der Variante "Großbritannien" ist jede Provinz Englands ein Versorgungszentrum und England starte mit sechs Armeen. Daher ist England die "stärkste" Nation, kann aber nichts tun, solange nicht eine andere Nation seine Armeen convoyen will (sonst muß England eine seiner Armeen abbauen "lassen", um nach der Wiedereinnahme des Zentrums eine Flotte aufbauen zu können).

Das Die Siegbedingung bleibt 18 Versorgungszentren.

Die Grafik der Standardkarte kann verwendet werden.

Chaos

Die Chaos-Variante wird mit 34 Spielern gespielt, wobei jeder ein einzelnes Heimatversorgungszentrum auf der Standardkarte hat. Das Spiel startet damit, daß die Spieler Aufbauten für den Winter 1900 abgeben.

Wichtig: Diese erste Runde wertet der GM am besten aus, indem er die Aufbauten direkt in GAME.TXT einträgt.

Das Aufbauen von Einheiten kann in jedem Zentrum geschehen, daß man besitzt (dafür gibt es die Pseudo-Nation "All" in FIELDS.TXT).

Wegen der hohen Spieleranzahl werden die Nationen wahlweise mit einer einbuchstabigen Abkürzung, oder mit dem Startfeld der Nation bezeichnet. Diese sind (Man beachte, daß "A"rmee und "F"lotte weggelassen wurden):

B Ank	H Bul	M Hol	R Mos	W Por	1 Smy	6 Tun
C Bel	I Con	N Kie	S Mun	X Rom	2 Spa	7 Ven
D Ber	J Den	O Lpl	T Nap	Y Rum	3 Pet	8 Vie
E Bre	K Edi	P Lon	U Nor	Z Ser	4 Swe	9 War
G Bud	L Gre	Q Mar	V Par	0 Seb	5 Tri	

Die Siegbedingung bleibt 18 Versorgungszentren.

Das Kartenmalen dürfte hoffnungslos sein. Die PlainText-Ausgabe funktioniert nicht.

Crowded

Diese Variante wird auf der Standardkarte gespielt, mit der Ausnahme, daß im Ruhrgebiet (Ruh) ein Versorgungszentrum ist. Jedes der 35 Versorgungszentren ist ein Heimatversorgungszentrum einer der 11 Nationen in dieser Variante. Die sieben Standardnationen starten mit ihren normalen Aufbauten. Zu ihnen hinzu kommen die vier neuen Nationen:

Balkan	(Ba):	F Gre,	F Rum,	A Ser,	A Bul
Lowland	(Lo):	F Bel,	A Hol,	A Ruh	
Norway	(No)	F Nor,	F Den,	A Swe	
Spain	(Sp):	F Por,	A Spa,	F Tun	

Das Die Siegbedingung bleibt 18 Versorgungszentren.

Die Grafik der Standardkarte kann zwar verwendet werden, doch existieren keine Einheitenbildchen für 11 Nationen \times 2 Einheitentypen. Außer dem 2 \times 7 Bildern sind nur noch 2 \times 2 weitere bei DipWork dabei (x1_a_64.tif, x1_f_64.tif und x2_a_64.tif, x2_f_64.tif). Das heißt man wartet mit dem Kartenmalen entweder bis zwei Nationen ausgeschieden sind, oder benutzt einige Zeichen doppelt (z.B. für Spain das gleiche wie für Russia). Eine Datei PICLIST.TXT muß man sich selber anlegen.

Pure

Dies ist eine einfache traditionelle Variante von Diplomacy. Es gibt die gewöhnlichen sieben Nationen. Auf dem Brett gibt es sieben Felder — eines zu jeder Nation: seinem Heimatversorgungszentrum. Die Felder sind alle miteinander verbunden — jedes mit jedem.

Zu Beginn hat jeder Spieler eine Armee in seinen Heim-SC. Das Die Siegbedingung sind 4 Versorgungszentren.

Das Kartenmalen ist hier nicht vorgesehen.

Seismic

Auswerten einer Partie Seismic-Diplomacy:

An sich verläuft eine Auswertung dieser Variante genauso, wie eine Standard-Partie, doch müssen zusätzlich noch die Geografiedateien FIELDS.DAT und BORDERS.DAT den Beben der Spieler entsprechend geändert werden. Dies geschieht mittels des Field-Converters DIP_FLDC. Eine genaue Beschreibung dieses Programms findet man an anderer Stelle. Eine kurze Zusammenfassung erhält man, wenn man das Programm mit dem Parameter "/" aufruft. Ich will hier das Wichtigste anmerken: Aufgrund der starken Verkomplizierung der Beben durch die Existenz von Subfeldern (Küsten), sollte man erwägen, ob man die Partie nicht ganz ohne Subfelder spielt. Die Datei VARIANTS\STANDARD\OHNESUBS.TXT ist dafür im Paket enthalten.

- Normal die Befehle für die Einheiten auswerten
 - Das Kartenmalen fällt weg, außer, man macht sich die Mühe, und malt mit einem geeigneten Programm, das ungepackte TIF-Grafiken im korrekten Format abspeichern kann, die Karte jedesmal neu und paßt dann auch die Koordinaten der veränderten Felder an.
 - Danach verändere man die Textdatei FIELDS.TXT den Beben entsprechend:
 - m Löschen der Grenzbeschreibungen der getrennten Felder. Dabei ist darauf zu achten, daß eine Grenze zweimal in der Datei vorkommt; einmal hin und einmal zurück.
 - m Hinzufügen von (je einer für hin und zurück) Grenzbeschreibungen der verbundenen Felder
 - m Sind Subfelder verschwunden? Entsprechende Informationen löschen: Feldinformationen und sämtliche (!) Grenzen, die damit zu tun haben.
 - m Sind Subfelder hinzugekommen? Entsprechende Informationen hinzufügen: Feldinformationen einfügen und Grenzen von diesem Feld und in dieses neue Feld hinein (!) einfügen. Uhrzeigersinn-Reihenfolge einhalten.
 - m Es ist wichtig, daß die Datei keine widersprüchlichen Feld- oder Grenzinformationen enthält. Sorgsam mehrfache Information löschen.
 - Den Fieldconverter aufrufen, bevor man die Auswertung zur nächsten Runde startet — am Besten, wenn man auch die letzte Auswertung "akzeptiert". Nach ein bißchen Rechnerei des Computers sind Dateien FIELDS.DAT und BORDERS.DAT den Beben in der Partie angepaßt worden. Hier wird sich übrigens eine Beschleunigung zeigen, wenn man ohne Subfelder spielt.
- Die Datei FIELDS.TXT, die im Verzeichnis VARIANTS\SEISMIC1 steht, ist die in der Antwort gespielte Partie "Japan" und hat schon einige Beben erlebt. Für eine eigene Partie muß man sich eine 'frische' Fields-Datei holen.

Anfügen einer weiteren Partie an das Projekt

Nun wird man ja nicht immer mit der beigelegten Beispielpartie herumspielen wollen, sondern man möchte seine eigene(n) Partie(n) in das Projekt einbauen. Wenn eine Partie neu startet ist das ein geringeres Problem. Da die Dateiverwaltung jedoch noch nicht implementiert ist, muß hier der Benutzer 'ran. Zu diesem Zweck ist ein Verzeichnis angelegt, daß eine 'leere' Standard-Diplomacy-Partie enthält. Die Zentren und die Standorte der Einheiten sind jene vom Winter 1900. Das Verzeichnis heißt 'ANY'. Um eine neue Standard-Partie zu installieren:

- Anlegen eines Verzeichnisses für die Partiedaten (z.B. "MYDIPPY") unter dem DipWork-Arbeitsverzeichnis
- Kopieren der Dateien GAME.TXT und ORDERS.TXT in aus "ANY" in das neu angelegte Verzeichnis oder Erstellen eigener Dateien.
- Editieren der Datei PROJECT.TXT (z.B. mit dem Hauptmenüpunkt) und unter GAMELIST den Pfad der neuen Partie hinschreiben ("mydip"). Man kann hinter einem Prozent "%" einen kleinen eigenen Hilfetext schreiben.
- fertig

Will man eine schon laufende Partie in das Projekt aufnehmen, ist GAME.TXT natürlich noch entsprechend zu ändern. In dieser Datei muß man jedoch darauf achten, daß man die Standard-Schreibweise benutzt, insbesondere ist die Großschreibung zu beachten, oder das man "BAS" und nicht "BAL" schreibt!

Installieren von Nicht-Standard-Partien

DipWork sind außer "Standard" auch einige andere Varianten beigelegt. Sie sind alle mit FIELDS.TXT, VARIANT.TXT, GAME.TXT und ORDERS.TXT ausgestattet. GAME.TXT enthält dabei jeweils die Standorte zu Beginn des Spiels. Angenommen, man möchte sich die Variante "Crowded" genauer ansehen oder eine Partie dieser Variante starten, dann sollte man sich dessen Beispielpartie verfügbar machen und folgendes tun:

- Ein Verzeichnisses unter dem DipWork-Arbeitsverzeichnis anlegen, das die Partie aufnehmen soll (z.B. "CROWDED")
- Kopieren der Dateien GAME.TXT und ORDERS.TXT in aus "VARIANTS\CROWDED" in das neu angelegte Verzeichnis (oder Erstellen eigener Dateien)
- Editieren der Datei PROJECT.TXT (z.B. mit dem Hauptmenüpunkt) und unter GAMELIST den Pfad der neuen Partie eintragen ("Crowded"). Man kann hinter einem Prozent "%" einen kleinen eigenen Hilfetext schreiben.
- Wenn es möglich ist, hier eine Karte zu malen (bei "Crowded" nicht möglich, weil für 11 Nationen nicht genug Einheiten-Bilder existieren), dann eine *.TIF-Datei im Variantenverzeichnis ablegen, z.B. die Datei VARIANTS\STANDARD\STANDARD.TIF kopieren.
- DIP_WORK starten und mit dem Menüpunkt "Service", "Geographie erzeugen" die Dateien FIELDS.DAT, BORDERS.DAT, SYNONYMS.DAT aus dem vorhandenen FIELDS.TXT schaffen
- fertig

Hinweise für Atari-Benutzer

Die Benutzer eines Ataris bekommen natürlich eine andere Programmversion, als jene, die einen PC besitzen. In einigen Punkten zeigt das Programm unterschiedliches Verhalten, oder man muß mit der Verwaltung anders umgehen. Da ich selber nur schlecht Zugriff auf einen Atari habe, haben andere Leute das Testen dahingehend übernehmen müssen. Was an Tips zusammengetragen wurde, wird hier erwähnt.

Systeminterpreter CLI.TOS

Weil das Atari-System standardmäßig keinen System-Interpreter eingebaut hat (wozu auch, normalerweise?), ist das Public-Domain-Programm "CLI.TOS" im Paket enthalten. Mit ihm ist das Anzeigen von Directories, das Kopieren von Dateien und ähnliches von DIP_WORK aus möglich. Allerdings fehlt eine detaillierte Dokumentation, da ich leider keine besitze -- wer hat eine?

Texteditoren, Alice

Das Aufrufen eines Texteditors mit einer Datei als Parameter, die nicht im aktuellen Pfad steht, bereitete unüberwindliche Schwierigkeiten. Die PC-User haben die Möglichkeit, vom DipWork-Menü aus, Textdateien zu editieren — für die Atari-Benutzer besteht diese Möglichkeit im Moment noch nicht. Ich erhoffe mir Besserung von dem PD-Editor Alice, der dem Paket beiliegt, den ich persönlich jedoch natürlich nicht ausprobieren kann. Für Alice gelten ähnliche Lizenzbestimmungen, wie für DipWork, genaueres ist in der Doku zu Alice nachzulesen. Ihr müßt euch, wie Johannes schreibt, davon überzeugen, daß Teile des Editors (das RSC-File) im Verzeichnis von DIP_WORK stehen! Eine neuere Version des Programms erfordert dies nicht mehr (ich glaube ab Version 2.0, schaut mal nach, welche Version dem Paket beiliegt)

Johannes Schwagereit dazu (27.8.92):

"Aufruf des Editor funktioniert. Getestet mit dem beiliegenden PD-Editor ALICE.PRG.

Beim Aufruf des Editors sind relative Pfadnamen zu vermeiden (z.B. SPAGIRIK\GAME.TXT), wir empfehlen absolute Pfadnamen, wie E:\DIP_WORK\SPAGIRIK\GAME.TXT.

Auch andere Editoren müßten funktionieren, Ausnahme ist aber EDISON.PRG. "

- Für den absoluten Pfadnamen existiert das vordefinierte Makro 'CURRENTPATH'
- Außer mit Edison hatte ich bei einem test auch mit Tempus Probleme.

"Can not evaluate, Can not eval, Can not, Can, C..."

Sollte der Scanner ständig "Can not evaluate this line" melden, kann das daran liegen, daß der Atari die Felder (FIELDS.DAT) und Grenzen (BORDERS.DAT) anders haben will als der PC. In dem Falle ist die Geografie-Textdatei (FIELDS.TXT) auf dem eigenen Rechner zu übersetzen. Dazu dient das Programm DIP_FLDC, das sich bei Aufruf ohne Parameter selbst erklärt. Aber es müßte wie folgt aufzurufen sein:

- DIP_FLDC VARIANTS\standard VARIANT.TXT FIELDS.TXT
- oder: DIP_FLDC VARIANTS\abstra2 VARIANT.TXT FIELDS.TXT
- oder: DIP_FLDC VARIANTS\seismic1 VARIANT.TXT FIELDS.TXT
- oder: DIP_FLDC VARIANTS\seismic1 VARIANT.TXT OHNESUBS.TXT

Hinweise von Michael Schröpl, 12.9.92:

TTP, PRG und TOS -- Was ist los? [nach Frage von Torsten]

Anmerkungen hierzu: [Auszug]

- *.TOS: Mauszeiger ausgeschaltet (für TOS-Programme). Programme, die aus DIP_WORK.??? heraus aufgerufen werden, sollten *.TOS heißen.
- *.TTP: Mauszeiger ausgeschaltet, nach Doppelklick erscheint eine Eingabebox, in der man commandline-Parameter eingeben kann (für TOS-Programme, die commandline-Parameter auswerten). Programme, bei denen man einen Parameter im Dialog angeben will bzw. muß (z. B. DIP_FLDC), sollten *.TTP heißen. (Hat man eine Kommando-Shell zur Verfügung, aus der heraus man Kommandos eingibt, dann reicht auch *.TOS.)
- *.PRG: Mauszeiger nicht ausgeschaltet(!) (für GEM-Programme). (GEM-) Programme, die eine Maus benötigen, müssen *.PRG (oder *.APP) heißen...

DIP_FLDC

Ich warne ausdrücklich davor, das Programm DIP_FLDC zu verwenden, ohne vorher nachzudenken! Das Programm sortiert eine Datei auf der Festplatte (oder dem Laufwerk...), nicht etwa im Hauptspeicher, noch dazu mit einem schlechten Algorithmus.

Anmerkung hierzu: Seit Version 2.0.4 nicht mehr! Zwar sortiert das Programm, um keinerlei Einschränkungen in der Felderanzahl zu haben immer noch halb-extern, aber enorm beschleunigt: Eine unsortierte Standard-Partie brauchte statt 1600 Vertauschungen nur noch 67! (Was mit Platten-Cache etwa doppelt so schnell ist, ohne habe ich es nicht probiert...

Kartenmalen

Das Kartenmalen selber läßt sich problemlos aufrufen. Jedoch ist mit dem Programm beim Eintragen von harmlosen 22 Start-Einheiten satte 5 Minuten auf der Platte die Hölle los.

PRINTTIF

Das Karten drucken ist auch auf dem einem Atari möglich. Es gibt jedoch auf dem Atari wenige Programme, die eine Konvertierung von TIF nach IMG vornehmen können. Leider steht mir kein solches zur Verfügung. (Neulich habe ich von einem gehört, mal sehen, ob das verfügbar ist!)

PRINTTIF ist eine Notlösung, um die theoretische Möglichkeit zum Dippy-Karten-Drucken zu schaffen. Wer TIF nach IMG umwandeln kann, soll das unbedingt tun und dann mit einem tollen Programm drucken. PRINTTIF ist nur sehr schlecht zur dauernden Benutzung geeignet — ich bin auf der Suche nach einer besseren Lösung.

Bei einem Test hatte das Programm bis Version 2.0.5 nach etwa 5 Minuten etwa 3% der Karte bearbeitet. Die nachfolgenden Versionen müßten um einen guten Faktor schneller sein (auf meinem Computer merke ich das nicht so...). Wichtig ist, daß eine Zieldatei angegeben wird, in die die Druckdatei ausgegeben werden soll, ansonsten hat man eine Datei "PRN" im DipWork-Verzeichnis. Diese Datei muß man dann ohne einen Filter oder ähnliches an den Drucker schicken (und dann am besten löschen). Wie das auf dem Atari geht, weiß ich nicht.

Immer noch Probleme?

Die Atari-Version kann ich nicht regelmäßig ausprobieren. Bitte ruft mich bei Schwierigkeiten an. Ich lerne gerne.

Most Frequently Asked Questions

001: Ich kann die Installation nicht durchführen, das Programm sagt "Disk full".

002: Wie kann ich Züge eingeben?

003: Beim Kartenmalen meldet das Programm "Can not find Mapfile". Was möchte es für eine Datei haben?

004: Wo steht eine Karte nach dem Malen?

005: Welche Dateien kann man nach einer Auswertung löschen?

006: Kann man die lästigen Tastendrucke zwischen den Auswertungsschritten abstellen?

007: Kann DipWork nur Zwei-Phasen-Partien auswerten?

Fragen und Antworten:

001: Ich kann die Installation nicht durchführen, das Programm sagt "Disk full".

Antwort: Rufe INSTALL von der Festplatte aus auf. Dazu solltest Du aber in dem Verzeichnis stehen, in dem Du DIPPY haben willst. Beispiel:

```
<C:\> md dippy
<C:\> cd dippy
<C:\DIPPY> copy a:*. *
<C:\DIPPY> install
<C:\DIPPY> del dippy20.exe
<C:\DIPPY> del demos.exe
```

002: Wie kann ich Züge eingeben?

Antwort: Die Eingaben erfolgen bei DipWork fast ausschließlich über eine Texteditor. Auf dem PC gibt es hunderte (oder mehr) frei kopierbare Texteditoren und alle mir bekannten Textverarbeitungsprogramme besitzen eine Option für "Speichern als reinen Text". Auf diese Weise nimmt man Veränderungen der Befehle (ORDERS.TXT) und aller anderen Dateien vor. In PROJECT.TXT ist der Lieblings-Texteditor mit vollem Pfad anzugeben. Beispiele:

```
EDITOR c:\dos\edlin.exe
EDITOR c:\nc\nce.exe
EDITOR c:\word5\word.exe
EDITOR c:\context\context.com
```

003: Beim Kartenmalen meldet das Programm "Can not find Mapfile".

Antwort: Es ist notwendig vor *jedem* Malen die leere Karte aus dem Varianten-Verzeichnis in das Partieverzeichnis zu kopieren! Diese Funktion ist standardmäßig im Menü eingebaut. Es gab auch schon mal Probleme, wenn das Archiv-Bit der Datei STANDARD.TIF nicht gesetzt war, was ich mir niemals erklären konnte. Mit dem DOS-Befehl

```
attrib +a <partie>\standard.tif
```

kann man das Archiv-Bit der Datei setzen.

004: Wo steht eine Karte nach dem Malen?

Antwort: Im Partieverzeichnis steht eine TIF-Grafik (bei Standardvariante STANDARD.TIF), die während des Malens verändert wird. Nach dem Malen ist die Karte aktuell.

005: Welche Dateien kann man nach einer Auswertung löschen?

Antwort: Wer Platz sparen will, der kann in einem Partieverzeichnis einer Standard-Partie einige Files löschen, wenn die Auswertung vollständig gemacht wurde — einschließlich Gestaltung und Ausdrucken. Nur folgende Dateien sollten übrigbleiben:

- NEWGAME.TXT
Enthält die Züge (mit evtl. Unterstreichungen), Standorte und Zentren nach der aktuellen Auswertung und ist vor der nächsten Auswertung in GAME.TXT umbenennen, bzw. zu kopieren.
- ORDERS.TXT
Nicht unbedingt nötig, es empfiehlt sich jedoch, die Datei nicht jedesmal ganz neu zu schreiben, sondern nur alte Befehle zu löschen und die neuen einzutragen.
- GAME.TXT
Wird nach korrekter Auswertung auch nicht mehr unbedingt benötigt (wird ja vor der nächsten Auswertung von NEWGAME.TXT überschrieben), aber es empfiehlt sich, die Datei aufzuheben, damit bei Beanstandungen von Seiten der Spieler die Auswertung wiederholt werden kann.

Also sind überflüssig:

ERRORS.TXT (gelöschte Befehle),

*.ORD (Zwischenstände)

*.TIF (Karte)

*.BAK (Sicherheitskopien angelegt von Deinem Texteditor...)

006: Kann man die lästigen Tastendrucke zwischen den Auswertungsschritten abstellen?

Antwort: Manche, die 25 Partie am Stück auswerten wollen, möchten so wenig wie möglich Kommunikation mit dem Computer betreiben. Andere, die vielleicht noch nicht ganz firm in der Programmbenutzung sind, oder bei der Zugeingabe öfter Tippfehler machen, wollen das Programm ganz schrittweise laufen lassen.

- Die wenigsten Tastendrücke werden bei der Einstellung "SLOWMOTION 0" in VARIANT.TXT verlangt. Den Autoren-Hinweis kann man nicht abschalten ;-). Außerdem wartet das Programm bei Programmabbruch durch Fehler auf eine Taste (seit V2.0.7).
- Die meisten Tastendrücke für ganz schrittweise Auswertung erreicht man mit der Einstellung "SLOWMOTION -1".
- Steht in der MENULIST am Ende einer Kommandozeile (die 3. Zeile) ein Ausrufezeichen in einfachen Hochkommas ('!'), dann wird nach der fehlerfreien Ausführung dieses Menüpunktes nicht auf einen Tastendruck gewartet.

007: Kann DipWork nur Zwei-Phasen-Partien auswerten?

Antwort: Aber mitnichten! Man muß nur die Schritte der Auswertung anders einteilen, also den "Winterer" noch nicht direkt nach dem Auswerten der Konflikte aufrufen. Die dann aktuellen Spielstände sind in `CONFCTED.ORD` (statt `wintered.ord`) gespeichert, diese Datei gibt man dann mit der PlainText-Ausgabe aus:

DIP_RUN partie /op=conflcted.ord

In späteren Programmversionen wird eine bessere Implementierung des Phasencharakters einer Diplomacy-Auswertung verwirklicht sein.

Syntax für einen scannbaren Diplomacy-Befehl

Ein "scannbarer" Diplomacy-Befehl ist ein vom Spieler angegebener Befehl für eine Einheit in ORDERS.TXT. Diese werden vom "Scanner" gelesen. Dann wird versucht, daraus ein computer-brauchbares Format zu erstellen (die *.ORD-Dateien).

Angegeben in 'BNF'. Nach DipWork V2.0 am Beispiel einer Standardvariante.

Field = 'Abc' | 'ABC'

/ Ein Field kann in seiner Standard- oder einer seiner Synonym-Schreibweisen angegeben sein */*

Unit = 'A' | 'F'

Nation = 'Au ' | 'En ' | 'Fr ' | 'Ge ' | 'It ' | 'Ru ' | 'Tu '

Location = [Nation] Unit Field

$$\text{SupportableOrder} = \text{MoveOrder} \mid \text{HoldOrder}$$

ConvoyableOrder = MoveOrder

```
SupportOrder = ' S ' Location SupportableOrder | ' S '
```

ConvoyOrder = ' C ' Location ConvoyableOrder

HoldOrder = ' xxx' | '-xxx'

MoveOrder = '-' Field

$$\text{AnyOrder} = \text{MoveOrder} \mid \text{HoldOrder} \mid \text{ConvoyOrder} \mid \text{SupportOrder}$$

Scanline = Location AnyOrder

Schematische Darstellung der Scannerauswertung

Nach dem hier abgebildeten Schema wertet der Scanner korrekte und fehlerhafte Befehle aus.

Die beiden Teile des Schemas sind nebeneinander zu fügen.

*** TELL I ***

```

+1-----+ +2-----+ +3-----+ +4-----+ +5-----+
|a Nation| |b else| +- F |a Field| |b else| +- B +- B
|b Unit| |a Unit| +-+ |a Field| +-+ |a Order| +-+ |c else| +-+
|c else| +- F |a Nation| +-+ |b Unit| +-+ |b " xxx| +- B +- A |a Nation| +->
|d " C| +-+ |b " S| +-+ |c " C| +-+ |b Unit| +->
|e else| +- B |d " S| +-+ |d " S| +-+
+-----+ +-----+ +-----+ +-----+ +-----+
*****
* Legende *
* * *
* A Move *
* B Hold *
* C Support to hold *
* D Support to move *

```

```

* E Convoyorder *|
* F Can't evaluate *|
*****|
-----+

*** TEIL II ***

+6-----+
|b else +- B +- B |b else +- B
| | |
->---+a Unit +-+ +15---^---+ +7-----+ |a Field+- E
+-----+ |b else | |a Order - +-----+
->---+-----+
|a Field +-+b else +- B
->--- Convoy +-----+ +-----+

+10-----+
|b else +- B +- B |b else +- B
| | |
->---+a Unit +-+ +16---^---+ +11-----+ |a Field+- D
+-----+ |b else | |a Order - +-----+
->---+-----+
|a Field +-+c else +- B
->--- Support +-----+
|b " xxx +- C
+-----+

```

Programmlimits

Maximale Beschränkungen für Werte

Alle Beschränkungen lassen sich auf Anfrage in einer besonderen Programmversion noch um einiges erhöhen, die absolut maximalen Werte sind in Klammern angegeben. Nur jene Limits, die mit einem (*) markiert sind, sind nur schwierig zu erhöhen.

- 255 Anzahl der zu verarbeitenden Befehle (>2000)
- 255 Anzahl der (auch leeren) Felder, die an Konflikten teilnehmen (*)
- 255 Länge einer Zeichenkette (*)
- 8 Einheitentypen (>100)
- 80 Zeilenlänge in VARIANT.TXT (255)
- 30 Anzahl der Nationen bei der PlainText-Ausgabe (*)
- 800 etwa, insgesamte Länge der Zentrennamen einer Nation (>5000)
- 90 Anzahl der Zentren einer Nation (>1000)
- 90 Anzahl der Grenzen eines Feldes (>1000)
- 80 Originalbefehl-Länge (255)
- 40 Feldnamenlänge (255)
- 16 Feldtypenlänge (255)
- 1024 Breite eines Einheitenbildes (>10000)

Erläuterungen zu Programmfunktionen, die keine "Bugs" sind

Einige Punkte des Programms können einiges Verwundern beim Benutzer auslösen. Zumindest einige werden hier ge- und erklärt — warum das so ist und nicht anders, ob es dem Programmierer überhaupt schon aufgefallen ist, daß... usw.

Gelöschte Befehle

Momentan werden die gelöschten Befehle (MULs, NSOs) noch nicht in die Auswertungsdatei geschrieben. Man kann sie aber ERRORS.TXT entnehmen.

Landesabkürzungen

Standard-Landesabkürzungen sind zweibuchstabig anzugeben:

Au En Fr Ge It Ru Tu

Einbuchstabige Abkürzungen kommen in Konflikt mit Einheitentypen, sind daher nicht möglich.

Presse-Befehl für Einheiten

Der Spielerbefehl "F NTH denkt" wird als IMP bemängelt, sofort in einen Haltebefehl umgewandelt und dann als "F NTH xxx?" in der Auswertung notiert. Naja, einige dumme Kommentare würden sonst schließlich nicht in die Zugtabelle passen (zum Beispiel: "Trolle Maramabad kauen dem GM ein Ohr ab..."). Darum findet man ihn als gelöschten Befehl mit vollem Text in ERRORS.TXT.

IMP Support

Der Spielerbefehl "F NTH S A Hol" wird in eine Unterstützung zum Halten umgewandelt und nicht, wie einige es wollten, in einen Haltebefehl. Meines Erachtens ist das korrekt, deshalb habe ich es so programmiert.

NSO ↔ IMP

Folgender falscher Spielerzug wird unkorrekt gekennzeichnet. Das Ergebnis ist jedoch das gleiche:

"A Mun-Kie; [F ENG C A Mun-Kie]?" sollte es sein,

"A Mun-Kie; [F ENG C A Mun-Kie]&" ist es.

Das liegt daran, daß bei der Flotte erstmal nur getestet wird, ob der Befehl gegeben wurde, der convoyt wird. Da aber auf Move-per-convoy getestet wird, wird "A Mun-Kie" nicht erkannt, was im Prinzip korrekt ist, weil so ein Zug kann ja nicht convoyt werden. Beides läuft jedoch auf einen Haltebefehl hinaus. Den Endeffekt erreicht die Art der Auswertung jedoch, folglich gibt's keinen Auswertungsfehler.

Automatischer Abbau

Bei Abbauten wird die Entfernung der abzubauenden Einheiten zum nächsten Heimat-SC nicht kalkuliert. Die Reihenfolge des Abbaus ist daher selbst zu bestimmen.

Befehligen fremder Einheiten

Man beachte, daß das Programm keine Möglichkeit hat, zu überprüfen, ob ein Spieler gerechtfertigterweise Befehle für Einheiten abgibt, die nicht zu seiner Nation gehören. Diese Kontrolle muß der GM übernehmen.

Verrutschungen in PLAIN.TXT

Ich habe gemerkt, daß die Tabellen REATREATS und BUILDUPS vertikal etwas verschoben sind. Theoretisch weiß ich woran es liegt, sieht aber nach viel Arbeit aus — "aber wir arbeiten daran..."

Interesse geweckt?

Aus gleichem Hause sind noch andere Programme erhältlich, die jedoch nur auf einem PC lauffähig sind, sich also nicht portieren lassen.

KAPDIP — Wertet eben jenes Spiels aus (Version 4.3)

Programm, das viel leistet, hat insbesondere eine völlig frei gestaltbare Form der Ausgabe. Diese ist direkt an den Drucker mit Unterstreichungen und Kursivstellungen möglich. Dokumentation ist vorhanden. Dieses Programm ist fertig und bedarf trotz einiger kleiner Bugs keiner Erweiterung mehr.

RENNEN — Macht die Rennen bei Turbo-2000 (Version 3.0)

Eingabe von Hand, oder bei erneuter Renndurchführung auch aus der Datei, die automatisch mitgeschrieben wurde. Ausgabe schon ordentlich formatiert in zwei Zeilen pro Fahrzeug. Und der spezielle Leckerbissen: Über jedes Fahrzeug wird eine Statistik erstellt, die einem bei der Ausschüttung der Extragelder hilft, oder es ermöglicht interessante Rennkommentare zu schreiben - normalerweise ein Manko der automatisierten Auswertung. Mit besonders ausführlichem Hilfesystem. Dieses Programm ist komplett fertig und mir fällt fast keine Verbesserungsmöglichkeit mehr ein. Anbei noch einige Turbo-Utilities für Erwartungswerte beim Rennen mit viel (d) und (r), sowie Erwartungswert und Übersicht für Entwickeln.