

GEMFONT

Ein GEMgestützter Zeichensatzeditor für GDOSPixelfonts

Version 1.11

Oktober 1992 bis Mai 1993

(c) 199193 by Sascha Blank

SharewareVersion

"Für moderne GEMProgramme eignet sich [...] GNUC [nicht]."
(STComputer 2/93, Seite 27)

Inhaltsverzeichnis

1. Rechtliches	5
1.1. Bedingungen für die Weitergabe von GEMFONT	5
1.2. Haftungsausschluß	5
1.3. Urheberrechte und deren Schutz	5
1.4. Bestimmungen zur Verbreitung selbsterstellter Fonts	6
2. Einleitung	7
2.1. Warum überhaupt GEMFONT?	7
2.2. Diese Anleitung	7
2.4. Systemanforderungen	8
2.5. Die Formalitäten	8
2.5.1. Wie bekomme ich GEMFONT?	8
2.5.2. Das SharewarePrinzip	9
2.5.3. Die Registrierung	9
2.5.4. Meine Gegenleistung	10
2.6. Sonstige Leistungen	10
2.6.1. Wie bekommt man ein Update?	10
2.6.2. Programmfehler	10
2.6.3. Probleme mit GEMFONT	10
2.6.4. Vorschläge	11
2.7. Austausch von Fonts	11
2.8. Danksagungen	11
3. Installation und Bedienung	12
3.1. Installation und Programmstart	12
3.2. Die GEMErweiterungen von GEMFONT	12
3.2.1. Wertgebundene Slider	12
3.2.2. Modale verschiebbare Dialoge	12
3.2.3. Über Tastatur bedienbare Dialogboxen	12
3.2.4. Menüzeilen in Fenster	13
3.2.5. RaveUpMenüs	13
3.2.6. Das Klemmbrett	13
3.3. Der Desktop	13
3.4. Das Editorfenster	13
3.4.1. Aufbau	13
3.4.2. Bedienung	14
3.5. Bedienung der Fenster über Funktionstasten	14
3.6. Sonstige hilfreiche Funktionen	15
4. Das GDOSFontformat	16
4.1. Einführung	16
4.2. Der Aufbau von Fonts	16
4.3. Der Fonthead	16

5.	Das Hauptmenü	24
5.1.	Das Menü "GEMFONT"	24
5.1.1.	"Über GEMFONT..." (Control Z)	24
5.2.	Das Menü "Datei"	24
5.2.1.	"Neu anlegen..." (Control N)	24
5.2.2.	"Öffnen..." (Control O)	24
5.2.3.	"Schließen..." (Control U)	25
5.2.4.	"Sichern" (Control S)	25
5.2.5.	"Sichern unter..." (Control M)	25
5.2.6.	"Sichern Metafont..." (Control H)	25
5.2.7.	"Abbrechen..." (Control D)	26
5.2.8.	"Entfernen..." (Control L)	26
5.2.9.	"Importieren..." (Control I)	26
5.2.10.	"Exportieren..."	27
5.2.11.	"Clipboard laden"	27
5.2.12.	"Clipboard speichern"	27
5.2.13.	"Clipboard löschen"	28
5.2.14.	"Ende" (Control Q)	28
5.3.	Das Menü "Block"	28
5.3.1.	"Blockanfang" (Control B)	28
5.3.2.	"Blockende" (Control K)	28
5.3.3.	"Alles auswählen" (Control A)	28
5.3.4.	"Ausschneiden" (Control X)	28
5.3.5.	"Kopieren" (Control C)	29
5.3.6.	"Einfügen" (Control V)	29
5.3.7.	"Blockkopiermodi..."	29
5.4.	Das Menü "Puffer"	29
5.4.1.	"Maske nach Puffer..." (Alternate P)	29
5.4.2.	"Puffer nach Maske..." (Alternate M)	29
5.4.3.	"Puffer speichern..." (Alternate U)	29
5.4.4.	"Puffer laden..." (Alternate O)	30
5.5.	Das Menü "Parameter..."	30
5.5.1.	"Speicherstatistik..."	30
5.5.2.	"Parameter einstellen..." (Alternate Q)	30
5.5.3.	"Parameter speichern..."	32
5.6.	Das Menü "Fonts"	32
5.6.1.	"Öffne xxxxxxxx.xxx" (Control 1 bis 7)	32
6.	Das Editorfenstermenü	33
6.1.	Das Menü "GEMFONT"	33
6.1.1.	"Über GEMFONT..."	33
6.2.	Das Menü "Zeichen"	33
6.2.1.	"Auswahl..." (Funktionstaste 1)	33
6.2.2.	"Breite" (Funktionstaste 2)	33
6.2.3.	"Hor. Verschiebung..." (Alternate V)	34
6.2.4.	"Zentrieren..." (Funktionstaste 7)	34
6.2.5.	"Ausmaß anpassen..." (Funktionstaste 8)	34
6.2.6.	"Werkzeuge..." (Funktionstaste 3)	34

6.2.7.	"Schieben/Rotieren..."	(Funktionstaste 4)	35
6.2.8.	"Spiegeln/Drehen..."	(Funktionstaste 5)	35
6.2.9.	"Stauchen/Strecken..."	(Funktionstaste 6)	35
6.2.10.	"Verfetten"		36
6.2.11.	"Komplett löschen"	(Alternate L)	36
6.2.12.	"Zeichen aufräumen"		36
6.3.	Das Menü "Editor"		36
6.3.1.	"Editormaskengröße..."	(Alternate C)	36
6.3.2.	"Editorrastergröße..."	(Alternate R)	37
6.3.3.	"Hilfspunkt setzen"	(Funktionstaste 9)	37
6.3.4.	"Hilfspunkt löschen"	(Alternate W)	37
6.3.5.	"Alle Punkte löschen"		37
6.3.6.	"Änderungen zurücknehmen"	(UNDO)	37
6.3.7.	"Undopuffer aktualisieren"	(Control UNDO)	37
6.4.	Das Menü "Fontheadere"		37
6.4.1.	"Zeichenextrema..."	(Alternate Z)	37
6.4.2.	"Fontlinien..."	(Alternate A)	38
6.4.3.	"Fontflags..."	(Alternate B)	38
6.4.4.	"Fontkenndaten..."	(Alternate D)	38
6.4.5.	"Kursivschrift..."	(Alternate K)	38
6.4.6.	"Fettschrift..."	(Alternate F)	38
6.4.7.	"Unterstreichen..."	(Alternate G)	38
6.4.8.	"Helle Schrift..."	(Alternate H)	38
6.5.	Schriftproben		38
7.	Häufige Fragen, Probleme und Tips		40
8.	Bezugsquellen für GDOS		41
8.1.	GDOS		41
8.2.	GDOSFonts		41
9.	Interna		43
9.1.	Wie alles anfang		43
9.2.	Ein paar Fakten		43
9.3.	Laden und Speichern von Fonts		43
10.	Fontregistrierung		45
10.1.	Die Idee		45
10.2.	Die Registrierung		45
10.3.	Anfragen		45

Rechtliches

Bedingungen für die Weitergabe von GEMFONT

GEMFONT wird als Shareware vertrieben. Es ist ausdrücklich erwünscht, daß GEMFONT weitergegeben wird. Dies darf jedoch nur in **vollständiger und unveränderter Form** geschehen, d.h. es müssen stets alle Dateien unverändert weitergegeben werden. Das Uploaden in Mailboxen, FTPServer und anderen Netzen ist ebenfalls ausdrücklich erwünscht, da ich selbst keinen direkten Zugang zu Mailboxnetzen habe. Es ist aber ausdrücklich verboten, GEMFONT kostenpflichtig im PDVersand zu vertreiben (Ausnahmen erfordern eine schriftliche Genehmigung meinerseits). Zuwiderhandlungen können rechtlich verfolgt werden! Ansonsten: Bitte empfehlen und verbreiten Sie GEMFONT weiter!

Haftungsausschluß

Obwohl GEMFONT mit großer Sorgfalt erstellt worden ist, kann ich nicht ausschließen, daß noch Fehler im Programm und/oder der Anleitung existieren. Deshalb stelle ich folgendes fest:

Die in dieser Dokumentation enthaltenen Informationen sind zum Zeitpunkt ihrer Veröffentlichung wahrscheinlich korrekt. Trotzdem werden alle Informationen ohne irgendeine Garantie weitergegeben.

Ich lehne jede Haftung für Schäden und sonstige Folgen ab, die direkt oder indirekt durch die Benutzung von GEMFONT und/oder der Dokumentation entstanden sind. Dies gilt auch für den Fall, daß die Schäden auf Fehler im Programm und/oder der Dokumentation zurückzuführen sind. Die Benutzung erfolgt stets ausschließlich auf eigene Gefahr. Ebenso garantiere ich nicht dafür, daß GEMFONT für einen bestimmten Zweck tauglich ist. Des weiteren lehne ich jede Haftung für Verstöße gegen das Urheberrecht ab, die in Zusammenhang mit dem nachfolgenden Kapitel "Urheberrechte und deren Schutz" stehen.

Sie dürfen GEMFONT für den Eigengebrauch modifizieren, z.B. Laufzeitpacken oder patchen (obwohl sich das nicht lohnt, da es weder einen Kopierschutz noch sonstige Lächerlichkeiten (etwa zeitgekoppelte Demophasen (die meisten sind eh' so primitiv implementiert)) zu entfernen gilt). Selbstverständlich übernehme ich erst recht keine Verantwortung für solche modifizierten Versionen. Zudem verbiete ich die Weitergabe solcher modifizierter Versionen an Dritte.

Urheberrechte und deren Schutz

Mit GEMFONT ist es u.a. möglich, Signum und BitmapFonts in GDOSFonts umzuwandeln. Bitte achten Sie dabei unbedingt auf den urheberrechtlichen Schutz der Fonts, die Sie konvertieren wollen.

Die konvertierten Fonts unterliegen den gleichen rechtlichen Bestimmungen wie die Originalfonts!

Unterliegen die zu konvertierenden Fonts keinem Urheberrecht bzw. hat der/die/die Autor/in/en das Recht auf die Weitergabe oder die Konvertierung der Fonts in andere Formate nicht weiter eingeschränkt, so spricht im allgemeinen nichts dagegen, diese nach der Konvertierung in GDOSFonts erneut in Umlauf zu bringen. Dies trifft auf viele SignumFonts zu, die meist den Status "Public Domain" tragen.

Daneben gibt es Fonts, die urheberrechtlich geschützt sind, beispielsweise die SignumFonts der "Professional FontDisks". In diesem Fall beziehen sich die Einschränkungen der Weitergabe sowohl auf die Originale als auch auf die daraus konvertierten Fonts. In der Regel dürfen diese Fonts nicht weitergegeben werden, so daß eine Konvertierung ins GDOSFormat **ausschließlich** für den eigenen Gebrauch erlaubt ist, deren Weitergabe an andere ist aber **nicht** erlaubt.

Bitte beachten Sie diesen Hinweis in Ihrem eigenen Interesse. Im Zweifelsfall sollten Sie das Einverständnis des/der/der Autor/in/en einholen oder zu Ihrem eigenen Schutz von einer Verbreitung absehen.

Bestimmungen zur Verbreitung selbsterstellter Fonts

Die Verbreitung von Fonts, die von Ihnen selbst erstellt wurden und ausschließlich ihrem eigenen Copyright unterliegen (also frei von Rechten Dritter sind), wird hiermit uneingeschränkt erlaubt. Es müssen weder Lizenzgebühren an den Programmautor gezahlt werden noch etwa muß um dessen Erlaubnis zur Verbreitung der Fonts gebeten werden. Dies gilt auch für den Fall, daß die Fonts als Teil eines kommerziellen Programmpakets verbreitet werden.

Die einzige Auflage, die ich stelle, ist diejenige, daß alle Produkte, die mit GEMFONT erstellte Fonts beinhalten, einen Hinweis auf GEMFONT tragen müssen. Dieser sollte beispielsweise im Handbuch, der Anleitung oder dem Programm selbst stehen und die Form

"Die diesem Programm beiliegenden GDOSFonts wurden mit dem Programm 'GEMFONT' von Sascha Blank, (c) 19911993 erstellt."

haben. Zudem muß erwähnt werden, wie man GEMFONT beim Programmautor beziehen kann, etwa so:

"Eine Kopie von GEMFONT erhalten Sie beim Autor von GEMFONT durch Zusendung eines frankierten Rückumschlags inkl. 3,5"Diskette an folgende Adresse: Sascha Blank, Brentanostraße 21, W5500 Trier" (Nach dem Inkrafttreten der neuen PLZen ist die alte PLZ durch die neue 54294 zu ersetzen.)

Wenn Sie eigene Fonts verbreiten wollen, so wäre ich dankbar, wenn Sie mir eine Kopie der Fonts zukommen lassen würden. Dies ist jedoch nicht zwingend nötig.

Wenn Sie weitere Fragen haben, dann wenden Sie sich bitte an mich.

Einleitung

Warum überhaupt GEMFONT?

"Jetzt habe ich endlich einen GDOSDruckertreiber, und dann gibt es keine Fonts dafür!"

Der Autor, im Sommer 1992

So ging es mir, als ich nach langem Suchen auf zahlreichen FTPServern endlich einen GDOSDruckertreiber für meinen NECPinwriter gefunden hatte. Einen Treiber hatte ich also nun, aber keine Fonts dafür, so daß ich über das VDI eigentlich keine Texte auf dem Drucker ausgeben konnte. Mit Hilfe eines FXEmulationsprogramms und der GDOSZeichensätzen für den Epson FX85 aus der Atari GDOSDistribution gelang mir es dann doch, aber die Qualität ist bei 9Nadlerfonts zwangsläufig enttäuschend.

Die weitere Suche nach Fonts verlief ergebnislos. Das Entwerfen eigener Fonts war kaum möglich, weil zum einen der eine Editor, den ich ebenfalls nach langem Suchen gefunden hatte, unbrauchbar war und zum anderen der Entwurf von Druckerfonts sehr aufwendig ist und ein gewisses Mindestmaß an typographischem Wissen erfordert, wenn das Ergebnis ansprechend sein soll.

Eine Möglichkeit wäre es, die vielen zum Teil sehr guten SignumFonts in GDOSFonts zu konvertieren, doch dies war bisher nicht möglich. (Angeblich gibt es ein Programm, das dies kann, doch es ist mir bis heute nicht begegnet.)

Um dem Fontnotstand zu mildern, beschloß ich, einen Fonteditor speziell für GDOSFonts zu entwerfen, der den Entwurf von Fonts möglichst einfach machen soll.

Inzwischen liegt das erste größere Update als Version 1.10 vor. Viele Ideen, die ich während meiner Vorbereitungszeit auf einige Klausuren gesammelt habe, sind hier mit eingegangen. Da sind vor allem die stark beschleunigten und erweiterten Vergrößer/Verkleinerungsroutinen sowie die Anzeige von VDITextattributen bei Schriftproben zu erwähnen.

Diese Anleitung

Diese Anleitung beschreibt (hoffentlich) alle Fakten, die nötig sind, um alle Funktionen von GEMFONT bedienen und deren Wirkung verstehen zu können. Dazu habe ich mich meist an das Prinzip "lieber lange und genau als zu kurz und zu knapp erklären" gehalten, denn umfangreiche Erläuterungen sind nötig, weil besonders Anfänger nicht über das zugegeben teilweise komplexe Insiderwissen verfügen; aber auch, weil GDOS und besonders seine

Interna bis heute nur spärlich dokumentiert sind und man viele Fakten selbst ermitteln muß.

Was GEMFONT so alles kann

GEMFONT bietet folgende Eigenschaften:

komplett in GEM eingebunden, dadurch lauffähig auf allen AtariSystemen in jeder Auflösung.

"zukunftsicher" in Bezug auf MultiGEM und MultiTOS durch verschiebbare modale Dialogboxen sowie durch die Nutzung des Klemmbretts.

Steigerung des Bedienungskomforts durch PopupMenüs, über Tastatur bedienbare Dialoge und Menüs, Menüzeilen in Fenstern, benutzerdefinierte Objekte (MultiTOSsauber) und einigem mehr. Bearbeitung von bis zu sieben Fonts gleichzeitig, die je in einemeigenen Fenster bearbeitet werden. (Diese Anzahl ließe sich durchaus noch steigern, doch mehr Fonts wären wohl nicht mehr sinnvoll.)

Import von Signum II Drucker und Editorfonts (9 und 24Nadel Format und 300 DPI Laserformat).

Import und Export von reinen BitmapFonts (Degas Format).

Laden und Speichern von Fonts im Motorola und Intelformat.

Speichern von MetafontInformationsfiles.

Teilweise Reparatur und Korrektur von defekten Fonts und solchen, die in falschen Formaten gespeichert wurden (nahezu alle GDOSFonts aus den Anfangszeiten von GDOS).

Manipulation aller Parameter im Zeichensatzheader möglich, einige davon werden auf Wunsch automatisch berechnet bzw. interpoliert.

zahlreiche Editorhilfen wie eingeblendete Ausrichtungslinien, frei wählbare Hilfspunkte.

zahlreiche Werkzeuge wie Linien, Rechtecke, Rahmen, halbe, viertel und ganze Kreise.

zahlreiche Effekte wie spiegeln, drehen, verschieben, rotieren, verfetten, stauchen, aushöhlen, umranden, invertieren, zentrieren etc.

Verkleinerung und Vergrößerung einzelner Zeichen oder ganzer Fonts. Seit Version 1.10 sorgen hochoptimierte Routinen für deutliche Geschwindigkeitszuwächse gegenüber Vorversionen (Faktor 10 bis 15).

Das editierte Zeichen wird ständig in Echtgröße angezeigt.

Eine beliebige Zeichenprobe kann auf dem Bildschirm zur Gesamtbeurteilung des Fonts ausgegeben werden. Sofern ein GDOSDruckertreiber vorhanden ist, kann ein Ausdruck auch auf einen Drucker in Echtgröße erfolgen. Seit Version 1.10 können

dabei auch die VDITextattribute hinzugeschaltet und kombiniert werden.

komfortable Puffer und Blockfunktionen.

Systemanforderungen

GEMFONT arbeitet mit jeder Bildschirmauflösung, jeder Farbfähigkeit, jedem Monitor und jeder Grafikkarte, sofern sie ordnungsgemäß durch das VDI unterstützt wird. Trotzdem sollte der Monitor wegen der Menüs eine Mindestauflösung von 80 mal 25 Zeichen bieten (muß aber nicht).

Der Betrieb ist auf allen Atari ST/STE/TT/FalconSystemen mit jeder beliebigen TOSVersion möglich (\geq TOS 1.04 sollte es aber schon sein). Da es unter MiNT keine Probleme gibt, sollte dies auf MultiTOS ebenfalls zutreffen. Der Betrieb unter MultiGEM sollte auch keine Probleme bereiten, was ich jedoch mangels Programm nicht selbst testen konnte.

Eine Festplatte ist nicht nötig, aber empfehlenswert. Zum Betrieb sollten mindestens 350 Kbyte freier Speicher vorhanden sein, da GEMFONT selbst etwa 280 Kbyte beansprucht. Doch beim Entwurf von großen Fonts, etwa für hochauflösende Drucker, sollte mindestens 1 Mbyte Gesamtspeicher zur Verfügung stehen, da so große Fonts im Editor leicht mehrere Hundert KByte in Anspruch nehmen (wohlgemerkt nur im Editor). Kurz gesagt: je mehr Speicher, um so besser. Die Größe des freien Speichers beeinflußt letztendlich die Maximalgröße der Fonts, die Sie bearbeiten können.

Ein GDOS ist nicht notwendig, außer, Sie wollen eine Schriftprobe auch auf dem Drucker begutachten. In diesem Fall benötigen Sie einen passenden GDOSDruckertreiber, aber keine passenden Fonts.

Die Formalitäten

Wie bekomme ich GEMFONT?

Eine Kopie von GEMFONT können Sie direkt bei mir erhalten. Dazu schicken Sie mir einfach eine 3,5"Diskette (DD, auf Wunsch auch HD) sowie einen an sich selbst adressierten und ausreichend frankierten Rückumschlag zu. Meine Adresse finden Sie weiter unten.

Das SharewarePrinzip

GEMFONT wird als Shareware betrieben. Für diejenigen, die

(immer) noch nicht wissen, was es damit auf sich hat, sei das Prinzip noch einmal kurz erklärt (siehe auch den Abschnitt "Rechtliches").

Shareware bedeutet, daß Sie beim Erhalt des Programms erst mal nichts zahlen müssen. Sie können dann über einen angemessenen Zeitraum (bis zu einem Monat ist angemessen) das Programm auf Herz und Nieren prüfen, ohne daß dabei Kosten entstehen. Wenn Ihnen das Programm nicht gefällt, dann vergessen Sie es am besten wieder. Sollte es Ihnen aber gefallen und/oder Sie benutzen es regelmäßig, dann wird die Sharewaregebühr fällig, die an der Programmautor zu zahlen ist (also an mich).

"Warum soll ich den überhaupt bezahlen?" werden Sie sich wohl fragen. Nun, in diesem Programm steckt viel Arbeit (inzwischen über zwei Jahre und alles in meiner Freizeit neben dem Studium), für die ich wenigstens eine kleine Entschädigung haben möchte, schließlich erhalten Sie ja auch ein professionelles Produkt.

Die Registrierung

Bei GEMFONT beträgt die SharewareGebühr 30, DM (ich nehme aber auch gerne mehr). Diesen Betrag können Sie mir auf verschiedene Arten zukommen lassen:

- 1.) als Bargeld oder
- 2.) in Form eines (Verrechnungs)Schecks

Nochmal meine Adresse:

Sascha Blank
Brentanostraße 21
W5500 Trier (Ab 1. Juli 1993: 54294 Trier)
Bundesrepublik Deutschland

Die Registrierung kann nur schriftlich erfolgen, füllen Sie dazu bitte das Formular am Ende dieser Anleitung aus und schicken Sie mir es zu.

Generell: wenn Sie mit mir Kontakt aufnehmen, dann bitte ich um ein wenig Geduld, sollte es einmal etwas dauern; das Studium geht halt vor. Wenn Sie sich schriftlich an mich wenden, legen Sie bitte ausreichend Rückporto bei (entfällt bei der Registrierung). Für Anwender mit Zugang zum InterNet bin ich auch über "blank@server.unitrier.de" erreichbar. Alternativ können Sie auch die Adresse "blank@treveris.unitrier.de" verwenden.

Ach ja: Dank dem neuen Konzept "Brief 2000" der Deutschen

Bundespost kostet der Versand einer Diskette in einer gepolsterten Versandtasche jetzt 3, statt bisher 1, DM. Da ich aber nicht einsehe, für diese Leistung jetzt dreimal soviel zu bezahlen wie bisher, zumal die Qualität des Versands nicht besser geworden ist, verschicke ich die Disketten ab sofort in verstärkten Briefumschlägen. Auf diese Weise kann ich das Porto je nach Gewicht der Beilagen auf 2, bzw. 1, DM drücken. Wer jedoch Angst hat, die Diskette könnte nicht heil bei Ihr/Ihm ankommen, der kann mir zusätzlich 3, DM in Briefmarken mitschicken und erhält dann Ihre/Seine Diskette wieder in einer gepolsterten Versandtasche.

Meine Gegenleistung

Wenn Sie mir die SharewareGebühr zukommen lassen, werden Sie bei mir als Anwender von GEMFONT registriert. Dies bringt ihnen folgende Vorteile:

Sie erhalten umgehend die neueste Version des Programms zugesandt.

Sie können jederzeit ein Update bei mir anfordern.

Wenn Sie ein Problem mit GEMFONT haben, dann können Sie sich jederzeit an mich wenden.

Wenn Sie einen Fehler gefunden haben, können Sie eine korrigierte Version von mir erhalten.

Sie erhalten neuere Versionen, die möglicherweise nur noch registrierten Anwendern zugänglich gemacht werden.

Sonstige Leistungen

Wie bekommt man ein Update?

Als registrierter Anwender können Sie jederzeit ein Update von GEMFONT bei mir erhalten. Dazu schicken Sie mir einfach eine 3,5"Diskette (DD, auf Wunsch auch HD) sowie einen an sich selbst adressierten und ausreichend frankierten Rückumschlag zu.

Sollten Sie nicht genau wissen, ob es bereits ein Update gibt, dann schicken Sie mir trotzdem eine Diskette und sagen mir zusätzlich, welche die neuste Version ist, die Sie besitzen.

Sollte diese Version noch die aktuelle sein, dann behalte ich Ihre Diskette einfach solange, bis es eine neuere Version gibt.

Diese schicke ich Ihnen dann automatisch zu, wenn sie veröffentlicht wird.

Programmfehler

Wenn Sie glauben, einen Fehler im Programm oder in der Anleitung gefunden zu haben, dann bitte ich Sie, mir diesen mitzuteilen. Dazu benötige ich eine genaue Beschreibung ihres Systems (Versionsnummer von GEMFONT, TOSVersion, Speicherausbau, Accessories etc.), je genauer, um so besser. Sollte der Fehler rekonstruierbar sein, so sagen Sie mir bitte, wie. Wenn Sie mir eine Diskette mitschicken (Modalitäten siehe oben), dann erhalten Sie eine fehlerbereinigte Version, sofern ich den Fehler finden und beheben kann. Tritt der Fehler in Zusammenhang mit bestimmten Fonts auf, dann ist es hilfreich, wenn Sie mir die Fonts auf die Diskette mitkopieren.

Probleme mit GEMFONT

Sollten Sie einmal ein Problem mit GEMFONT haben und es ohne anderweitige Hilfe nicht lösen können, dann können Sie sich an mich wenden. Ich werde mich im Rahmen meiner Möglichkeiten bemühen, Ihnen zu helfen. Auch hier gilt: Je genauer Sie mir Ihr Problem und die Umstände beschreiben, desto eher kann ich Ihnen helfen.

Vorschläge

Wenn Sie besondere Wünsche, Erweiterungen oder Vorschläge haben, die Sie gerne in zukünftigen Versionen von GEMFONT realisiert sehen würden, dann teilen Sie mir diese mit. Ich werde ich mich dann bemühen, diese aufzugreifen und nach eingehender Prüfung in die Tat umzusetzen.

Austausch von Fonts

Wenn Sie neue Fonts entworfen bzw. konvertiert haben und glauben, sie könnten auch für andere Benutzer nützlich sein, dann schicken Sie mir diese doch zu, sofern sich dies mit den rechtlichen Bestimmungen vereinbaren läßt (siehe Kapitel 1). Ich werde diese Fonts dann zusammen mit GEMFONT verbreiten, soweit es die Umstände, d.h. der Platz auf der Diskette zuläßt. Vielleicht entsteht so eine kleine (oder auch große) Sammlung an hochwertigen GDOSFonts. Besonders dringend werden "Brotschriften" gebraucht, also solche Schriften, die sehr häufig benutzt werden (beispielsweise Times oder Helvetica). Zier oder Frakturschriften sind zwar schön anzuschauen, werden aber nur selten genutzt.

Danksagungen

Bedanken möchte ich mich bei

der Free Software Foundation für ihren hervorragenden CCompiler und all die anderen Utilities, die unter ihrer Feder entstehen, sowie all denjenigen, die sie immer wieder auf TOS/MiNT portieren.

Eric R. Smith, Nick Castellano, Jhawar Bammi und allen anderen für ihre ausgezeichneten CBibliotheken und ihre Mitarbeit an MiNT.

dem Autorentrio Jankowski/Rabich/Reschke für ihr Profibuch (wann kommt die Fortsetzung?)

dem STMagazin für sein Expertenforum, ohne daß ich so manches Mal auf dem Schlauch gestanden hätte.

Joachim Höhne aus Osnabrück, der sich als erster bei mir registrieren ließ. Selbstverständlich möchte ich mich auch bei allen anderen registrierten Anwendern für ihre Ehrlichkeit danken.

Nicht bedanken möchte ich mich bei

dem, der das deutsche Handbuch zu Lattice C verfasst hat. (Wenn ich so die Leserbriefe in den wichtigen STPublikationen oder die Nachrichten im NewsNet lese, dann weiß ich, daß ich mit meiner Meinung nicht alleine stehe!)

dem, der meinen Account auf dem alten SUNServer aufgelöst hat.

(das kann eigentlich nur einer gewesen sein...)

einem gewissen Herrn für eine gewisse "Algorithmen und Datenstrukturen"Klausur im SS 92 (er wird wissen, daß er gemeint ist...)

Installation und Bedienung

Installation und Programmstart

Die Installation von GEMFONT gestaltet sich einfach. Kopieren Sie die Dateien GEMFONT.APP, GEMFONT.RSC und GEMFONT.INF (sofern vorhanden) an einen gleichen, aber beliebigen Platz, der Ihnen zusagt.

GEMFONT kann als GEMApplikation nicht aus dem AUTOOrdner gestartet werden. Sollten Sie es versehentlich doch einmal versuchen, dann bemerkt GEMFONT dies und verhindert Schlimmeres. Eine Installation als Accessory durch Umbenennen von GEMFONT.APP in GEMFONT.ACC ist zwar möglich, doch GEMFONT unterstützt den Accessorymodus nicht weiter, u.a. aufgrund der komplexen internen Speicherverwaltung. In diesem Fall belegt GEMFONT zwar einen Accessoryeintrag, doch der Versuch, dieses Accessory zu aktivieren, wird ignoriert.

Werden GEMFONT Fonts als Parameter übergeben (etwa durch Ziehen eines Fonts auf das GEMFONTIcon unter Newdesk oder Gemini), so werden diese Fonts automatisch geladen.

Die GEMErweiterungen von GEMFONT

GEMFONT benutzt neben den üblichen GEMElementen noch weitere, um den Benutzerkomfort weiter zu steigern.

Wertgebundene Slider

Wer XCONTROL kennt, der kennt auch die Slider, bei deren Bewegung irgendeine Anzeige gleichzeitig aktualisiert wird. Auch GEMFONT benutzt solche Slider. Ihre Bedienung funktioniert genau wie die FensterSlider.

Modale verschiebbare Dialoge

Modale verschiebbare Dialoge gehören inzwischen zum guten Ton eines GEMProgramms. Das bedeutet nichts anderes, als daß alle Dialoge in Fenstern liegen. Dadurch wird die Parallelverarbeitung von Prozessen auch während der Dialogbearbeitung nicht unterbrochen. Wird ein Dialog bearbeitet, so können alle anderen Fenster, die zur gleichen Applikation gehören, nicht nach oben gebracht werden. Eine Weiterarbeit ist erst dann möglich, wenn der Dialog beendet wurde. Zudem sind alle Menüeinträge gesperrt, andere

Accessories können jedoch weiterhin aktiviert werden. So kann die Applikation im Hintergrund weiterarbeiten, während ein anderer Prozess im Vordergrund läuft. Sollte kein Fenster mehr für die Anzeige eines Dialogs zur Verfügung stehen, dann wird er normal, also ohne Fenster bearbeitet. Die Parallelverarbeitung ist in diesem Fall natürlich unterbrochen. Sollte Ihnen das zu kompliziert sein, so brauchen Sie sich nur zu merken, daß alle Dialoge in Fenstern liegen und daß sich an deren Bedienung nichts geändert hat gegenüber normalen Dialogen.

Über Tastatur bedienbare Dialogboxen

Alle Dialoge sind über Tastatur bedienbar. Wenn in einem Button ein Zeichen unterstrichen ist, so können Sie diesen durch <Alternate>+<unterstrichenes Zeichen> aktivieren. Die Taste <UNDO> löst meistens einen Button aus, der sinngemäß für den Abbruch einer Aktion steht ("Ende", "Abbruch" etc.)

Menüzeilen in Fenster

Fenster können ebenfalls Menüzeilen besitzen. Um ein Menü auszuklappen, klicken Sie den Titel an, das Menü klappt je nach freiem Platz nach unten oder nach oben aus. Jetzt können Sie wie gewohnt einen Eintrag selektieren, ein Klick außerhalb der Menübox selektiert keinen Eintrag und bricht die Auswahl zudem ab. Halten Sie die Maustaste nach dem Klick auf den Titel gedrückt, dann klappt das Menü aus. Fahren Sie mit der Maus auf den gewünschten Eintrag. Dieser wird dann beim Loslassen der Maustaste selektiert (am besten mal ausprobieren). Auf diese Weise können Sie ein Menü mit einem Klick bedienen. Nach einer Selektion bzw. Nichtselektion (ein Klick außerhalb der Menübox) klappt das Menü in jedem Fall ein (hört sich schlimmer an, als es ist). Fenstermenüs können ebenso wie die echten Menüs über Tastaturkürzel gesteuert werden. Diese finden sie jeweils am rechten Rand des Eintrags.

Wenn ein Menü ausgeklappt ist und der Mauszeiger auf einem Eintrag steht, Sie aber dann das Kürzel eines anderen Eintrags tippen, so hat das eingegebene Kürzel Vorrang vor dem Eintrag, auf dem der Mauszeiger steht.

RaveUpMenüs

RaveUpMenüs (eigentlich heißen sie ja PopUpMenüs) werden genauso wie die Menüzeilen in den Fenstern bedient. Mit dem

kreisförmigen Pfeil rechts neben dem Menü kann nacheinander jeder Eintrag ausgewählt werden, ohne daß dabei das Menü ausgeklappt werden muß.

Klappt ein RaveUpMenü aus, so ist der bisher aktuelle Eintrag mit einem Häkchen gekennzeichnet. Bei eingeklapptem Menü wird jeweils der aktuelle Eintrag angezeigt.

Das Klemmbrett

Das Klemmbrett soll den Datenaustausch zwischen Programmen erleichtern, was besonders bei multitaskingfähigen GEMVersionen praktisch ist. Das Klemmbrett ist nichts anderes als ein Ordner auf einem Laufwerk (meist dem, von dem gebootet worden ist). Jedes Programm, das das Klemmbrett unterstützt, kann in das Klemmbrett hineinspeichern bzw. von ihm lesen. Die Namen der Dateien im Klemmbrett sind vom System vorgegeben, deshalb erfolgt auch keine Frage nach einem Dateinamen beim Laden oder Speichern. So kann ein Programm A eine Datei ins Klemmbrett ablegen und Programm B dann sofort danach die Datei aus dem Klemmbrett laden, ohne daß dabei irgendwelche Angaben gemacht werden müßten.

Der Desktop

Seit Version 1.03 besitzt GEMFONT kein eigenes Desktop mehr. Zu diesem Schritt entschloß ich mich, da das Desktop von GEMFONT nichts bot, was nicht auch über das Menü zu erreichen gewesen wäre. Zudem ist ein eigenes Desktop unter MultiGEM und MultiTOS nur hinderlich, wenn es nicht mindestens genauso viel bietet wie das StandardDesktop (und das tat es bisher eben nicht).

Das Editorfenster

Aufbau

Das Editorfenster besteht aus mehreren Komponenten. Der größte Teil nimmt das Editorfeld selbst ein. Im rechten Rand wird das jeweils editierte Zeichen in Echtgröße dargestellt. Die Titelzeile zeigt den kompletten Dateinamen des Fonts an. In der Infozeile des Fensters wird die aktuelle Position des Mauszeigers über der Editorgitter angezeigt sowie die Nummer des gerade bearbeiteten Zeichens. Befindet sich die Maus außerhalb des Gitters, dann werden keine Koordinaten angezeigt. Ein Stern am Anfang der Infozeile zeigt Ihnen, daß Sie den Font seit dem letzten Speicher/Ladevorgang verändert haben.

Unterhalb der Infozeile befindet sich eine Menüzeile, deren Bedienung bereits besprochen wurde. Der Pfeil im unteren Editorrand zeigt auf diejenige Pixelspalte, die die letzte rechte benutzte Zeichenspalte darstellt. Mit anderen Worten: dieser Pfeil gibt die Breite an. Der rechte Rand neben dem vertikalen Slider zeigt die Position der Ausrichtungslinien an. Dabei stehen folgende Buchstaben für folgende Linien:

T	Zeichenzellenoberkante ("Top Line")
A	Zeichenoberkante ("Ascent Line")
H	Halblinie ("Half Line")
G	Grundlinie
D	Zeichenunterkante ("Descent Line")
B	Zeichenzellenunterkante ("Bottom Line")

Das eigentliche Editorraster wird durch ein Gitter aus gestrichelten Linien gekennzeichnet. Zur besseren Orientierung sind alle Linien, die eine Ausrichtungslinie darstellen, von durchgezogenen Linien umgeben. Ebenso wird die rechte Linie derjenigen Spalte, die die Breite darstellt, mit einer durchgezogenen Linie gekennzeichnet.

Bedienung

Gesetzte Punkte werden durch ein schwarzes Kästchen im Gitter dargestellt, ungesetzte durch ein weißes Kästchen. Sollte ein Block markiert sein, so werden dessen vier Eckpunkte durch ein Diagonalkreuz in den jeweiligen Kästchen angezeigt. Punkte, die einen Hilfspunkt darstellen, werden grau gerastert angezeigt. Punkte werden innerhalb des Editorrasters mit der linken Maustaste gesetzt bzw. gelöscht. Dabei wird bei einem Klick der bisherige Zustand des Punktes umgekehrt. Das heißt ein Klick auf einen gelöschten Punkt setzt diesen Punkt und umgekehrt. Sie können mit gedrückter Maustaste gleichzeitig mehrere Punkte bearbeiten. Solange Sie die Maustaste drücken, werden alle Punkte, die sie mit der Maus berühren, verändert. Ob gelöscht oder gesetzt wird, hängt davon ab, welchen Zustand derjenige Punkt unterhalb des Mausepfeils hatte, als die Maustaste zu drücken begann. (Am besten Sie probieren das mal aus, dann verstehen Sie alles viel leichter.)

Bedienung der Fenster über Funktionstasten

Mit den Funktionstasten können Sie bestimmte Fensteranordnungen erreichen, die mit der Maus alleine nur schwer zu erreichen

wären:

(Shift F1)

Damit machen Sie ein Fenster nach dem anderen zum aktuellen. Die gleiche Funktion läßt sich über die Standardkombination (Control W) erreichen.

(Shift F2) Das aktuelle Fenster wird auf die kleinsten erlaubten Maße verkleinert. Dies ist praktisch, wenn man einmal vor lauter Fenstern den Desktop nicht mehr sieht. Mit dem FullSizeIcon des Fensters vergrößern Sie es wieder auf seine vorherigen Ausmaße.

(Shift F3)

Alle Fenster werden untereinander angeordnet, so daß sie die maximale Breite einnehmen und alle die gleiche Höhe haben. Sollte bei diesem Vorhaben jedoch ein Fenster zu klein oder zu schmal werden, so erscheint eine Meldung und die Fenster bleiben unverändert.

(Shift F4)

Alle Fenster werden nebeneinander angeordnet, so daß sie die maximale Höhe einnehmen und alle die gleiche Breite haben. Sollte bei diesem Vorhaben jedoch ein Fenster zu klein oder zu schmal werden, so erscheint eine Meldung und die Fenster bleiben unverändert.

(Shift F5)

Alle Fenster werden übereinander angeordnet, so daß ihre Arbeitsfläche möglichst groß ist, jedoch jedes andere Fenster direkt ohne Verschieben nach oben gebracht werden kann. Sollte bei diesem Vorhaben jedoch ein Fenster zu klein oder zu schmal werden, so erscheint eine Meldung und die Fenster bleiben unverändert.

Mit den Cursortasten kann der Fensterinhalt zeilen bzw. spaltenweise verschoben werden, wird zusätzlich CONTROL gedrückt, wird Seitenweise in die gewünschte Richtung verschoben.

Sonstige hilfreiche Funktionen

Sollte einmal aufgrund von RedrawFehlern der Inhalt eines Fensters zerstört worden sein, so genügt ein Druck auf die TabulatorTaste und das aktuelle Fenster wird komplett neu gezeichnet. Halten Sie zudem die ShiftTaste gedrückt, so wird

der Desktop neugezeichnet, was aber nur in schlimmeren Fällen nötig sein sollte.

Das GDOSFontformat

Einführung

Bei den Fonts, die GDOS verwendet, handelt es sich um PixelFonts, d.h. in einer Zeichenmaske sind Punkte gesetzt bzw. gelöscht, die dann als ganzes das Aussehen jedes Zeichens bestimmen.

Der Vorteil von Pixelfonts ist, daß sie bei "kleinen" Größen besser aussehen, dafür bei "großen" Größen viel mehr Speicher verbrauchen. Zudem ist ihre algorithmische Handhabung recht einfach.

Der Aufbau von Fonts

GDOS verwendet ein spezielles Format, in dem die Fonts abgelegt werden. Grob gesehen besteht ein Font auf maximal vier Komponenten:

- 1.)Der Fonthead. In ihm stehen alle Angaben, die das VDI zum Umgang mit dem Font braucht, angefangen vom Zeichensatznamen bis zu den Kursivüberhangsneigungen.
- 2.)Die Horizontale OffsetTabelle. Sie ist optional und wird nur sehr selten verwendet. Mit ihr kann jedes Zeichen vor seiner Ausgabe um einen frei bestimmbaren Wert nach links verschoben werden.
- 3.)Die Zeichenbreiten OffsetTabelle. In ihr ist vermerkt, wie breit jedes Zeichen ist. Dies ist besonders bei proportionalen Schriften wichtig.
- 4.)Die Zeichendaten. Dies sind die eigentlichen Daten, die das Aussehen des Fonts bestimmen.

Die Reihenfolge der Teile 2, 3 und 4 ist zwar so nicht zwingend vorgegeben, hat sich aber so eingebürgert.

Jedes Fontfile enthält nur **einen** Font in einer bestimmten Größe. Soll ein Font aus mehreren Größen bestehen, so wird für jede Größe ein eigener Font angelegt, wobei alle Fonts die gleichen Einträge *id* und *name* besitzen.

Der Fonthead

Der Fonthead enthält alle wichtigen Informationen, deren Kenntnis auch bei der Bedienung von GEMFONT wichtig ist.

Nachfolgend werden der Header und alle seine Einträge erklärt. Hinter der Deklaration steht in Klammern der dezimale Offset zum

Dateianfang in Bytes. Für alle NichtCProgrammierer seien kurz die verwendeten Datentypen erklärt:

```
char           = 128 bis 128 (1 Byte)
unsigned char  = 0 bis 255 (1 Byte)
short          = 32768 bis 32767 (2 Bytes)
unsigned short = 0 bis 65535 (2 Bytes)
```

Steht vor dem Variablennamen ein Stern "*", dann bedeutet dies "Zeiger auf Datentyp ..."

Noch ein Hinweis: Keine normale VDIAnwendung braucht sich jemals mit dem Aufbau und dem Format von GDOSFonts zu beschäftigen. Alle Einträge können über entsprechende VDIFunktion erfragt werden. Diese Beschreibung ist nur für das Editieren von GDOSFonts wichtig!

```
typedef struct font_hdr{
```

```
unsigned short id;                (0)
```

Jeder Font verfügt über eine Kennung, an der er eindeutig identifizierbar ist. Diese Kennung liegt im Bereich von 0 bis 32767. Das VDI erkennt seine Fonts ausschließlich an diesem Feld, nicht etwa am Fontnamen (**name**) oder am Filenamen des Fonts, wie manchmal fälschlicherweise behauptet wird. Manche Programme haben mit **IDs** größer als 255 oder 127 Schwierigkeiten. Dies liegt *nicht* an GDOS, sondern ist ein Fehler dieser Programme!

```
unsigned short point;            (2)
```

Damit wird die Größe des Fonts in typographischen Punkten angegeben (1pt = ca. 0.0351 cm). Zwischen welchen Linien dabei gemessen wird, ist allerdings nirgends dokumentiert und auch mir noch nicht ganz klar. GEMFONT ermittelt die Größe zwischen *Grundlinie* und *Zeichenoberkante*, weil diese Werte denen aus vielen anderen Fonts meist sehr ähnlich sind.

```
char name[32];                    (4)
```

Jeder Font besitzt einen Namen. Dieser wird zwar nicht zwingend vom VDI benötigt, kann aber ganz legal erfragt werden und sollte alleine schon deshalb immer angegeben werden. Der Name kann

maximal 32 Zeichen lang sein. Dabei ist man übereingekommen, daß die Zeichen 0 bis 15 den Namen des Fonts enthalten (etwa "Times", "Swiss") und die Zeichen 16 bis 31 den Stil (etwa "bold" oder "italic") enthalten. Der String ist nicht zwangsläufig NULterminiert.

```
unsigned short first_ade;           (36)
unsigned short last_ade;           (38)
```

Der ASCIIZeichensatz besteht auch 256 Zeichen, nämlich von 0 bis 255 inklusive. VDI ist sogar in der Lage, Fonts mit 65536 Zeichen (!) zu verarbeiten, was aber nicht gemacht wird, weil alle Hochsprachenbindings nicht darauf vorbereitet sind und ein solcher Font immens viel Speicher verbrauchen würde.

Deshalb enthält heute jeder GDOSFont maximal 256 Zeichen. Um noch mehr Speicherplatz zu sparen, kann ein Font auch weniger Zeichen enthalten. Zu diesem Zweck kann der Code des ersten und des letzten benutzten Zeichens angegeben werden. Eine im englischen Sprachraum beliebte Einschränkung sind Fonts, die nur die Zeichen 32 bis 127 inklusive enthalten. Dann fehlen zwar die ControlCodes sowie alle nichtenglischen Zeichen, doch dies stört in diesem Sprachraum nicht. Auf diese Weise werden die Verwaltungstabellen kleiner. In der Fontdatei werden nur die tatsächlich benutzten Zeichen abgelegt. Sollte ein Programm dennoch versuchen, ein Zeichen auszugeben, das nicht im Font vorhanden ist, so wird stattdessen ein Fragezeichen ausgegeben.

```
unsigned short top;                 (40)
unsigned short ascent;             (42) unsigned short half;
      (44)
unsigned short descent;           (46)
unsigned short bottom;            (48)
```

Jeder Font wird mit Hilfe von typographischen Hilfslinien erzeugt. Das VDI benutzt dazu sechs horizontale Ausrichtungslinien.

Zur Veranschaulichung stellen sie sich ein Schreibheft vor, welches Schulanfänger benutzen. Dort werden in der Regel Zeilen verwendet, die aus vier Linien bestehen. Die zweite Zeile von unten ist die Grundlinie (Baseline), die die Unterkante der meisten Buchstaben darstellt. Die direkt darüberliegende Zeile ist die Halblinie (Halfline), die die Oberkante vieler Kleinbuchstaben darstellt (etwa a, c, e, o). Die oberste Linie ist die Zeichenoberkante (Ascent Line), die wie der Name schon

sagt die Oberkante der meisten Großbuchstaben darstellt. Schließlich ist die unterste Linie die Zeichenunterkante (Descent Line). Sie wird nur von wenigen Buchstaben, etwa dem 'g' oder dem 'y' berührt. Damit wären also schon einmal vier der sechs Zeilen erklärt.

Bei der Textausgabe von Zeilen, die untereinander stehen sollen, geht man meist so vor, daß die Zeichenoberkante jeder neuen Zeile nahtlos an die Zeichenunterkante der vorherigen Zeile anschließt. Dabei kann es aber passieren, daß bestimmte Zeichen beider Zeilen sich optisch schneiden. Um dies zu verhindern, benutzt das VDI zwei weitere Linien. Die Zeichenzellenunterkante (Bottom Line) liegt ein wenig unterhalb der Zeichenunterkante, so wie die Zeichenzellenoberkante (Top Line) ein wenig über der Zeichenoberkante liegt. Sie bilden damit eine Art Sicherheitsabstand gegen Überschneidungen. Die Zeichenzellenoberkante liegt dabei immer auf der Position 0. Die Position der Linien ist jedoch nicht absolut, sondern relativ zur Grundlinie gespeichert. **top** gibt den Abstand der Grundlinie zur Zeichenzellenoberkante an. Alle anderen Linienpositionen sind relativ zu **top** angegeben. Ihre absolute Positionen errechnen sich wie folgt, die Numerierung beginnt bei 0:

```
Zeichenzellenoberkante = 0;
Zeichenoberkante       = top ascent;
Halblinie              = top half;
Grundlinie             = top;
Zeichenunterkante     = top + descent;
Zeichenzellenunterkante = top + bottom;
```

```
unsigned short max_char_width;    (50)
unsigned short max_cell_width;    (52)
```

Diese beiden Felder geben Auskunft über das breiteste Zeichen im gesamten Zeichensatz. **max_cell_width** gibt an, wie breit dieses Zeichen laut der Editoreinstellung ist. **max_char_width** hingegen gibt dessen echte Breite an. Die Breiten schließen den linken freien Randbereich mit ein!

Dazu ein Beispiel: Schauen Sie sich den Buchstaben "i" in ihrem Systemzeichensatz an. Laut Eintrag in der Zeichenbreitentabelle ist er 8 Pixel breit, doch sie sehen, daß er eigentlich schmaler ist. Wenn dieses "i" der breiteste Buchstabe wäre, dann enthielte **max_cell_width** den Wert 8. Subtrahiert man den Abstand, der zwischen dem "i" und seiner rechten Kante liegt, dann erhält man die echte Zeichenbreite, **max_char_width** enthielte dann beispielsweise den Wert 6.

```
unsigned short left_offset;          (54)
unsigned short right_offset;        (56)
```

Wenn ein Zeichen geneigt ausgegeben wird, dann ragt es überhalb der Grundlinie rechts über seinen Rand, unterhalb der Grundlinie über seinen linken Rand. Wie weit es nun herausragt, muß das VDI genau wissen, damit die Textausgabe auch bei geneigter Schrift halbwegs vernünftig aussieht.

Wie man diese Werte ermittelt, wurde nie offiziell dokumentiert. Eigene Versuche lassen aber auf folgendes Verfahren schließen:

```
|*****|
|*****|
|*****|
|*****| < Grundlinie
|*****|
|*****|
```

Die Striche um das Zeichen sollen die Zellenkanten andeuten, die **nicht** mehr zum Zeichen gehören. Die Sterne stehen für alle diejenigen Pixel, die gesetzt werden können, ob sie es tatsächlich sind, spielt bei der Kursivausgabe keine Rolle. Wenn dieses Zeichen nun geneigt wird, sieht es wie folgt aus:

```
|      *****
|      *****
|      *****
|*****| < Grundlinie
***** |
***** |
```

Sie sehen, das Zeichen ragt an beiden Seiten über die Zellenränder hinaus. Unterhalb der Grundlinie ragt es um zwei Pixel nach links über (der Zellenrand gehört nicht mehr zum Zeichen), über der Grundlinie ragt es drei Pixel nach rechts. In diesem Fall würde **left_offset** den Wert 2, **right_offset** den Wert 3 enthalten. (Wahnsinn!)

Dieses Verfahren ist zweifellos nicht das einfachste, erfordert aber keine aufwendigen mathematischen Berechnungen (etwa trigonometrische Funktionen) und ist deshalb recht günstig zu implementieren.

unsigned short **thicken**; (58)

Soll ein Zeichen fett dargestellt werden, so wird das Zeichen ausgegeben, und anschließend noch einmal um genau ein Pixel nach rechts versetzt erneut ausgegeben. **thicken** gibt dabei an, wie oft dieser Vorgang wiederholt werden soll. Man sollte mit diesem Wert sparsam umgehen, meist reicht der Wert 1, bei großen Fonts auch der Wert 2, da sonst das Zeichen besonders auf dem Bildschirm so sehr "verschmiert", daß es unleserlich wird.

unsigned short **ul_size**; (60)

Beim Unterstreichen wird eine Linie gezogen, die auf der Ausrichtungslinie "Zeichenunterkante" liegt. Dieser Wert gibt an, wie dick diese Linie ist. Auch hier reichen die Werte 1 oder 2 meist vollkommen aus. Das VDI legt die Oberkante des Unterstrichs (ich weiß nicht, wie man das richtig nennt) genau auf die Zeichenzellenunterkante und zeichnet dann eine **ul_size** Pixel dicke Linie, die nach unten hin in wächst.

unsigned short **lighten**; (62)

Um ein Zeichen "hell" darzustellen, wird ebenfalls ein recht komplizierter, aber einfach zu implementierender Algorithmus benutzt. Wenn Sie ihn nicht verstehen sollten, dann macht das auch nichts.

Zur aufgehellten Zeichenausgabe wird eine Bitmaske benutzt, mit der die Zeichendaten ständig verknüpft wird. Dazu wird von der Zeichenzellenoberkante ausgehend die erste Zeile mit **lighten** UNDverknüpft und das Ergebnis dargestellt. Dann wird **lighten** um ein Bit nach links rotiert und die nächste Zeile mit diesem Wert UNDverknüpft. Dies geht solange, bis die Zeichenzellenunterkante erreicht worden ist.

Die Voreinstellung ist der Wert 0x5555, der bei einem schwarzen Zeichen ein Schachbrettmuster ergibt. Dieser Wert paßt immer und ergibt auch den besten Effekt.

Nur wenn der Wert **lighten** aus einem regelmäßigen Bitmuster besteht, ergeben sich "gutausschende" helle Zeichen. Bei sehr kleinen Zeichen kann es dennoch passieren, daß es nicht mehr zu erkennen ist, doch dies ist prinzipbedingt.

unsigned short **skew**; (64)

Hier wird es noch komplizierter als bei **lighten**. Diese Maske bestimmt, wie sehr ein Zeichen bei der Kursivschrift nach rechts geneigt sind. Das verwendete Verfahren, das ebenfalls nicht dokumentiert ist, ermöglicht Neigungen zwischen 0 und 45 Grad nach rechts.

Die Berechnung beginnt bei der Zeichenzellenunterkante. Ist Bit 0 von **skew** gesetzt, dann wird die auszugebende Zeile um ein Pixel nach rechts verschoben, bevor sie ausgegeben wird. Jetzt wird **skew** um ein Bit nach rechts rotiert und die Zeile über der vorherigen Zeile bearbeitet. Ist Bit 0 von **skew** gesetzt, so wird diese Zeile um ein Pixel gegenüber der *vorherigen* Zeile versetzt ausgegeben. Das ganze wiederholt sich, bis alle Zeilen ausgegeben sind. (So so!)

Auch hierbei ergeben nur gleichmäßige Bitmuster in **skew** eine saubere Neigung, der Standardwert 0x5555 ist immer eine gute Wahl, er ergibt eine Neigung von 22,5 Grad. Die maximale Neigung von 45 Grad kann mit dem Wert 0xFFFF erreicht werden, der Wert 0x0000 erzeugt keine Neigung.

unsigned short **flags;** (66)

Zur Zeit sind in diesem Wert nur die Bits 0 bis 3 benutzt:

Bit 0 wird intern vom VDI verwendet. Ist es gesetzt, dann gibt es an, ob ein Zeichensatz momentan der aktuelle ist (durch `vst_font` ausgewählt).

Ist Bit 1 gesetzt, dann wird die Horizontale Offset Tabelle vom Font benutzt.

Ist Bit 2 gesetzt, dann liegt der Font im MotorolaFormat vor. Da GEM einst auf dem PC entstanden ist, wurden auch einige dort herrschende Konventionen mit auf den Atari übernommen. Auf einem PC liegen unter anderem das High und LowByte bei 16BitWörtern im Gegensatz zum Atari genau vertauscht vor; dies heißt allgemein "IntelFormat". Wenn ein Font im IntelFormat vorliegt, muß er erst einmal ins MotorolaFormat umgewandelt werden, damit das AtariGEM wieder etwas mit ihm anfangen kann. Damit GDOS nun weiß, in welchem Format der Font vorliegt, fragt es dieses Bit ab.

Bit 3 zeigt im gesetzten Zustand an, daß alle Zeichen im Font die gleiche Breite besitzen, also nicht proportional sind.

Alle anderen Bits sind reserviert für Erweiterungen und müssen immer auf 0 gesetzt sein.

Mit Bit 2 hat es so einiges auf sich. Schon bei der Abfrage dieses Bits beginnen die Probleme. Liegt der Font im MotorolaFormat vor, so liegt Bit 2 von `flags` im Byte 67 relativ

zum Dateianfang. Bei einer Abfrage von Bit 2 wird also auf Byte 67 zugegriffen. Liegt der Font aber im IntelFormat vor, dann sind ja High und LowByte vertauscht, Bit 2 liegt jetzt im Byte 66 (in Worten: sechshundsechzig). Um dieses Bit nun abzufragen, müßte man bereits vorher wissen, in welchem Format der Font vorliegt. Nur unter der Annahme, daß Bit 10 des Flags niemals benutzt wird, ist eine sichere Abfrage möglich. Aber abgesehen davon kümmert sich kaum ein Fonteditor um dieses Bit. So sind mir bereits Fonts begegnet, die laut *flags* im IntelFormat vorliegen, tatsächlich aber im MotorolaFormat abgelegt sind, beispielsweise der Font "42nd street". Andere Fonts mischen einfach Intel und MotorolaFormat. Als Beispiel sei der Font "Gemini" (Filename: GEMINI.FNT; wenn ich bloß noch wüßte, woher ich den hatte), der anscheinend in mehreren Versionen im Umlauf ist, genannt: Laut *flags* MotorolaFormat, die Fontdaten ebenfalls im MotorolaFormat, die Zeichenbreitentabelle dafür aber im IntelFormat. Die Dokumentationen (wenn man das so nennen darf) zu GDOS schreiben vor, daß die Zeichendaten im gleichen Format vorliegen müssen wie der Font selbst, doch dies ist (zumindest für den Atari) falsch (oder alle meine Versuchsfonts sind falsch). Die Fontdaten liegen *immer* im MotorolaFormat vor, egal, ob der Rest des Fonts im IntelFormat abgelegt ist oder nicht. Versuche bestätigen diese Annahme. Nichtsdestotrotz gibt es einige Fonts, die es besser zu wissen glauben. Fazit: Auf dieses Bit war bisher kaum Verlaß. Deshalb benutzt GEMFONT komplizierte Plausibilitätsalgorithmen, mit denen das echte Format der Tabellen ermittelt wird, um möglichst unabhängig vom FormatBit zu sein. Lediglich beim Format der Fontdaten muß manchmal manuell nachgeholfen werden, sollte einmal ein Font geladen werden, dessen Zeichendaten tatsächlich im IntelFormat vorliegen. Es versteht sich von selbst, daß GEMFONT beim Speichern auf ein korrektes Format der Fonts achtet.

```
unsigned char *hor_table;          (68)
```

Jedes Zeichen kann vor seiner Ausgabe noch einmal um eine bestimmte Anzahl von Pixeln nach rechts verschoben werden. Dazu kann in dieser Tabelle jedem Zeichen ein Wert zwischen 0 und 255 zugeordnet werden. Bisher nutzt kein mir bekannter Font diese Möglichkeit, wahrscheinlich, weil es bisher keinen Fonteditor gab, der dies konnte.

Der Sinn dieser Funktion scheint darin zu liegen, ein einfachesZeichenkerning (Unterschneiden von zwei benachbarten Zeichen) zu realisieren. Um dies zu verdeutlichen, sehen Sie

sich die Buchstabenkombination "Te" an. Besonders bei großen Zeichen klafft zwischen dem 'T' und dem 'e' eine unansehnliche Lücke, die dadurch geschlossen werden könnte, indem man das 'e' näher an das 'T' heranschiebt, Platz genug ist ja da. Bei GDOS kann jedoch nur das linke Zeichen um einen freien Wert nach rechts verschoben werden, als würde das 'T' an das 'e' herangeschoben werden. Beide Zeichen überschneiden sich zwar, ohne das es jedoch dabei zur Auslöschung eines Teils des rechten Zeichens kommt.

Mit dieser Methode läßt sich aber nur ein einfaches Kerning realisieren, denn das Kerning ist bei den Buchstaben 'Te' ein anderer Wert als bei 'Tr' oder 'TW' etwa, und genau dies läßt sich mit dieser Methode der konstanten Rechtsverschiebung nicht verwirklichen. GDOS wählt stets für das gerade auszugebende Zeichen ein festes Kerning, ohne dabei erst einen Blick auf das nächste Zeichen zu werfen und dann in Abhängigkeit beider Zeichen das Kerning auszuwählen, so wie es sein sollte.

Es scheint zudem vom jeweils aktuellen Treiber (oder gar GDOS?) abhängig zu sein, ob tatsächlich der horizontale Offset auch berücksichtigt wird. Bei eigenen Versuchen unter NVDI waren jedenfalls keine Veränderungen zu erkennen.

Wird keine Tabelle im Font genutzt, so zeigt dieser Zeiger auf den Anfang der nächsten benutzten Komponente (siehe auch Kapitel 4.2). Dieser und die beiden nachfolgenden Zeiger geben die Position immer relativ zum Dateistart an.

Diese Tabelle enthält insgesamt (**last_ade first_ade + 1**) Einträge, sofern sie denn benutzt wird. Der zu dem Zeichencode x gehörende Eintrag findet sich in (x **first_ade**).ten Position in der Tabelle.

Beispiel: **last_ade** = 32, Zeichen = "2" (Code 50), die Position des dazugehörigen Zeichens befindet sich dann an der (50 32 = 18).ten Position in der Tabelle.

Wichtig: Die Numerierung der Einträge beginnt bei 0, nicht etwa bei 1.

```
unsigned short *off_table;          (72)
```

Hier wird verzeichnet, wie breit jedes Zeichen ist. Dabei wird jedoch nicht einfach in jeden Eintrag die Breite eingetragen. Stattdessen wird die XKoordinate, bei der ein Zeichen im "Fontbild" anfängt (siehe Beschreibung **dat_table**) eingetragen, denn dies vereinfacht VDI die Ausgabe mittels den Rasterkopierfunktionen erheblich. Die echte Breite eines Zeichens errechnet sich dann aus der Differenz zwischen der Startposition des gesuchten Zeichens und der Startposition des unmittelbar nachfolgenden Zeichens im "Fontbild". Daraus ergibt

sich, daß diese Tabelle einen Eintrag mehr enthält als Zeichen benutzt werden.

Beispiel: Breite des Zeichens mit dem Code x ergibt sich aus ((Wert an Position (x + 1 **first_ade**)) (Wert an Position (x **first_ade**))).

```
unsigned short *dat_table;          (76)
```

Dieser Zeiger zeigt auf die eigentlichen Fontdaten. Erklärung folgt. Der Begriff "Fontdaten" wird in der Anleitung für diesen Teil des Fonts gebraucht.

```
unsigned short form_width;          (80)  
unsigned short form_height;        (82)
```

Stellen sie sich alle Zeichen eines Fonts hintereinander auf einen riesigen Bildschirm ausgegeben vor. Genauso sind die Fontdaten im Font abgelegt, wie in einem Bild. **form_width** gibt dabei die Breite des Bildes in Bytes an (nicht in Words wie manche behaupten). **form_height** gibt die Höhe des Bildes in Pixeln an. Die Breite beträgt immer ein Vielfaches von 2 Bytes, auch wenn dies nicht dokumentiert ist. Sollte das Bild nicht genau auf einer Bytegrenze enden, dann werden soviele leere Pixelspalten an das Fontbild angehängt, bis die Grenzen stimmen.

```
struct font_hdr *next_font;        (84)
```

Dieser Zeiger wird intern vom VDI benutzt. Die Fonts werden als lineare Liste organisiert, und dieser Zeiger zeigt auf den nächsten Font in der Liste. In der Fontdatei sollte er immer auf 0 gesetzt sind.

```
} FONT_HDR;
```

Das Hauptmenü

Viele Dialogboxen besitzen einen Button mit dem Text "Abbruch", "Ende" oder ähnlich. Damit kann die ausgewählte Operation abgebrochen werden, so daß der gleiche Zustand herrscht wie vor dem Aufruf des Dialogs. Da dies bei vielen Dialogen der Fall ist, wird dies nicht mehr gesondert erwähnt.

Entsprechend den GEMKonventionen können viele Menüpunkte auch per Tastatur aktiviert werden, die betreffenden Tastenkombination sind in Klammern angegeben. Drei Punkte hinter einem Menütitel bedeuten, daß nach der Aktivierung noch mindestens eine Dialogbox folgt, bevor die eigentliche Aktion ausgelöst wird.

Das Menü "GEMFONT"

"Über GEMFONT..." (Control Z)

Hier finden sie allgemeine Informationen über GEMFONT, unter anderem die Versionsnummer und das Erstellungsdatum der Version, die sie benutzen. Inoffizielle Versionen enthalten zur besseren Unterscheidung einen kleinen Buchstaben direkt hinter der Nummer, doch werden solche Versionen normalerweise nicht von mir in Umlauf gebracht.

Bei Rückfragen bitte immer Versionsnummer und datum angeben.

Das Menü "Datei"

"Neu anlegen..." (Control N)

Mit dieser Funktion können Sie einen neuen Font anlegen. sofern nicht bereits sieben Fonts bearbeitet werden.

GEMFONT fragt Sie nach der Größe des größten Zeichens innerhalb des Fonts. Diese Angaben dienen zum Einrichten der Editorpuffer für jedes Zeichen, damit nur soviel Speicherplatz verbraucht wird, wie auch (wahrscheinlich) benötigt wird. Erfahrungsgemäß sollten Sie hier etwas großzügig sein. Sollten Sie sich doch mal verschätzt haben, so können Sie die Größe später nachträglich ändern.

"Öffnen..." (Control O)

Sie laden hiermit einen GDOSFont, den Sie in der Dateiauswahlbox auswählen können. Existiert der von Ihnen gewählte Font nicht

oder sollte es sich dabei nicht um einen GDOSFont handeln, dann werden Sie darauf hingewiesen und die Funktion wird abgebrochen. GEMFONT versucht so weit wie möglich zu kontrollieren, ob es sich bei der ausgewählten Datei tatsächlich um einen GDOSFont handelt. Da GDOSFonts aber keine MagicNumbers besitzen, ist die Überprüfung nicht komplett sicher, zumal einige ähnliche Zeichensatzformate ebenfalls die Endung "FNT" besitzen. Deshalb sollten Sie GEMFONT keine Dateien unterschieben, die offensichtlich keine GDOSFonts sind.

Da man dem Filenamen allgemein nur schwer ansehen kann, welcher Font dahintersteckt, wird eine Informationsbox angezeigt, der Sie die wichtigsten Informationen des Fonts entnehmen können. Dazu gehören der Name, die Größe in Punkten, die Breite des breitesten Zeichens sowie die Fontflags. In der Box "Konvertierung" stellen Sie ein, ob die Zeichendaten konvertiert werden sollen. Dies ist erfahrungsgemäß aber nicht nötig, weshalb Sie diese Option nur in Ausnahmefällen aktivieren sollten. Sagt Ihnen der Font zu, dann können Sie in laden, ansonsten die Funktion abbrechen.

"Schließen..." (Control U)

Es wird das aktuelle Editorfenster geschlossen.

"Sichern" (Control S)

Hier können Sie einen Font im GDOSFormat abspeichern. Falls der Font noch keinen Namen besitzt das ist der Fall, wenn er neu angelegt oder importiert wurde, dann werden Sie nach dem Dateinamen gefragt, unter dem der Font abgelegt werden soll. Ansonsten wird der Name benutzt, der in der Titelzeile des dazugehörigen Fensters steht.

Sollte GEMFONT feststellen, daß die Ausrichtungslinien noch nicht positioniert worden sind, dann werden Sie darauf hingewiesen und der Vorgang abgebrochen. Dies ist deshalb wichtig, weil die Menge der zu speichernden Fontdaten durch die Position Zeichenzellenunterkante vorgegeben ist. Das heißt im Klartext, das alle Pixelinformationen, die sich unterhalb oder Zeichenzellenunterkante oder rechts von der von Ihnen eingestellten Breite befinden, **nicht** mitgespeichert werden. Und wenn die Zeichenzellenunterkante noch auf Position 0 liegt, dann gehen alle Informationen verloren (ist mir selbst einige Male passiert, deshalb diese Vorsichtsmaßnahme).

Bei der Ermittlung der echten Zeichenbreiten wird der Parameter **max_char_width** korrekt gesetzt. Anschließend werden die Daten zum Fontbild zusammengesetzt. Die Flags unter "Fontflags"

bestimmen dabei das Format der Daten. Dabei wird im Gegensatz zu einigen anderen GDOSFonteditoren auf das korrekte Format aller Tabellen geachtet.

Sollte aus irgendeinem Grund ein Fehler beim Speichern auftreten, so werden Sie darauf hingewiesen und der Vorgang abgebrochen. Die fehlerhafte Datei wird automatisch gelöscht. Noch ein Hinweis: GEMFONT setzt den Font vor dem Speichern fix und fertig im Speicher zusammen, bevor er auf ein externes Medium kopiert wird. Dafür wird Speicherplatz gebraucht, der vorübergehend angefordert wird. Deshalb sollten Sie stets darauf achten, nicht den kompletten Speicher zu belegen, sonst hat GEMFONT keinen Platz mehr für den Aufbau. Bei der Erstellung von Druckerfonts sollten mindestens 100 KByte frei sein (siehe "Speicherstatistik").

Dieses Verfahren wurde gewählt, um auch in Multitaskingumgebungen flexibel zu sein, wo sich ja dauernd die Speicherlandschaft ändert. Das ständige Belegen eines Aufbauspeichers, der nur für kurze Zeit genutzt wird, würde anderen Applikationen nur Speicherplatz wegnehmen, den Sie vielleicht selbst besser brauchen könnten. Dies bringt aber nur etwas, wenn auch die anderen Prozesse so kollegial sind.

"Sichern unter..." (Control M)

Hier geschieht das gleiche wie unter "Speichern", doch können Sie einen neuen Namen bestimmen, unter dem der Font abgelegt werden soll. Nach dem erfolgreichen Speichern werden Sie gefragt, ob Sie den eben ausgewählten Dateinamen übernehmen oder den alten beibehalten wollen.

"Sichern Metafont..." (Control H)

Das Format von GDOSMetafontdateien ist bis heute nicht dokumentiert worden. Eigene Untersuchungen an den wenigen Metfontdateien, die es gibt, legen die Vermutung nahe, daß es sich dabei um GDOSDruckerfonts handelt, denen aber die eigentlichen Fontdaten fehlen; sie enthalten nur die eigentlichen Verwaltungsdaten.

Wofür die Metafonts gut sein sollen, scheint auch niemand so genau zu wissen, jedenfalls ist sich die Literatur nicht einig, ob man überhaupt Metafonts braucht oder nicht. Wenn GDOS für die Bearbeitung von Metafile Angaben über die benutzten Fonts braucht, dann könnte es sich die auch direkt aus den Druckerfonts selbst beschaffen (denke ich mir zumindest). Wer genaueres über den Aufbau und Sinn der Metafonts weiß, dem

wäre ich dankbar, wenn er sich mit mir in Verbindung setzen würde, damit ich vielleicht diese Funktion korrigieren kann. Kurz noch zur Funktion selbst: Nach der Angabe eines Dateinamens wird der momentan bearbeitete Font gespeichert, außer, daß das Fontbild in dieser Datei fehlt.

"Abbrechen..." (Control D)

Wollen Sie alle gemachten Änderungen verwerfen, so können Sie mit diesem Menüpunkt den aktuellen Font erneut einladen. Dies funktioniert aber nicht, wenn der Font neu angelegt oder importiert wurde.

"Entfernen..." (Control L)

Es wird der Font, zu dem das aktuelle Editorfenster gehört, aus dem Speicher entfernt und sein Platz wieder freigegeben. Wurde der Font geändert, so erfolgt vorher eine Sicherheitsabfrage.

"Importieren..." (Control I)

Diese Funktion ist eines der Glanzlichter von GEMFONT. Hier können Sie Signum IIFonts (Editor und Druckerfonts) als auch BitmapFonts importieren. Damit eröffnet sich GDOS eine Vielzahl von teilweise professionellen Fonts aus der SignumWelt. Da Signum wohl jedem Benutzer ein Begriff ist, erkläre ich nur kurz, was BitmapFonts sind: BitmapFonts haben eine konstante Zeichengröße von 8x8 oder 8x16 Pixeln (Dateilänge 2048 oder 4096 Bytes), sie enthalten ausschließlich die Fontdaten aller 256 Zeichen. Der Editor "Tempus" oder die Textverarbeitung "Protex" können diese Zeichensätze benutzen, um den Systemzeichensatz innerhalb dieser Programme zu ersetzen. BitmapFonts spielen für GDOS keine große Rolle, wurden jedoch ihrer Einfachheit halber implementiert.

Nach der Anwahl dieser Funktion können Sie das Format angeben, das Sie importieren wollen. Sie haben die Auswahl zwischen SignumEditorfonts, SignumDruckerfonts (egal ob *.L30 oder *.P24) und den BitmapFonts in zwei Größen. Haben Sie sich entschieden, dann wählen Sie die entsprechende Datei aus. GEMFONT überprüft, ob die Datei das gewünschte Format besitzt und reklamiert Ihre Wahl, wenn dies nicht der Fall ist.

Bei SignumFonts wird die maximale Größe der Zeichen ermittelt. Die dabei ermittelten Werte werden Ihnen als neue Größe der Editorpuffer vorgeschlagen. Diese Werte können Sie in den

meisten Fällen übernehmen. Geben Sie kleinere Werte an, dann gehen die Daten, die außerhalb liegen, verloren. Wollen Sie den Font vergrößern, dann geben Sie dementsprechend größere Werte an.

Bei manchen Fonts kommt es vor, daß alle Zeichen recht tief unterhalb der Zeichenzellenoberkannte liegen. GEMFONT fragt Sie deshalb, ob Sie alle Zeichen um einen intern ermittelten Wert höhersetzen wollen. Da dieses Verfahren nicht immer funktioniert, sollten Sie den Font zuerst unversetzt laden und erst dann entscheiden, ob Sie ihn erneut versetzt importieren wollen. Bevor Sie mit der Arbeit beginnen, kontrollieren Sie unbedingt alle diejenigen Zeichen, die besonders nahe an der oberen oder unteren Zeichenzellenkante enden, denn dort kann dann manchmal ein Stück verlorengegangen sein.

Da Signum IIFonts nur aus 127 Zeichen bestehen, die zudem in einer anderen Reihenfolge als dem ASCIIFormat abgelegt sind, verteilt GEMFONT die Zeichen so, daß sie mit möglichst vielen ASCIIZeichen übereinstimmen. Probleme gibt es dabei jedoch nur mit denjenigen Zeichen, die unter Signum dem externen Ziffernblock zugeordnet werden können, dafür gibt es keine äquivalenten ASCIIPositionen. Folgende Tabelle zeigt genau, wohin welches SignumZeichen gelegt wird:

S i g n u m C o d e											
		x0	x1	x2	x3	x4	x5	x6	x7	x8	x9
	+										
	0x	0	194	195	196	197	198	199	200	201	202
	1x	203	204	205	206	207	208	209	210	211	16
A	2x	17	18	19	20	21	22	23	24	25	212
S	3x	213	214	221	33	34	35	36	37	38	39
C	4x	40	41	42	43	44	45	46	47	48	49
I	5x	50	51	52	53	54	55	56	57	58	59
I	6x	60	61	62	63	129	65	66	67	68	69
	7x	70	71	72	73	74	75	76	77	78	79
C	8x	80	81	82	83	84	85	86	87	88	89
o	9x	90	148	154	132	94	95	96	97	98	99
d	10x	100	101	102	103	104	105	106	107	108	109
e	11x	110	111	112	113	114	115	116	117	118	119
	12x	120	121	122	153	124	142	126	158		

Leider müssen viele SignumFonts noch per Hand nachbearbeitet werden, häufig wegen der Zeichenbreiten, die bei manchen Fonts im Trunkenheitszustand gesetzt worden zu sein scheinen. Doch auch hierfür bietet GEMFONT passende Werkzeuge.

(Da ich selbst weder Signum noch Script besitzt, fehlt mir auch

die offizielle Dokumentation der SignumFontformate, die vielleicht in den Handbüchern zu diesen Programmen abgedruckt ist. Die jetzigen Routinen basieren auf einem Quicktip, der in einer Ausgabe der STComputer erschienen ist. Für weitere Hinweise bezüglich der Formate wäre ich sehr dankbar). Der Import der BitmapFonts gestaltet sich problemlos, es sind dafür keine weiteren Angaben durch den Anwender notwendig.

"Exportieren..."

GEMFONT kann Fonts auch wieder im BitmapFormat exportieren (für SignumFonts ist schließlich Skarabus zuständig). Dabei wählen Sie wieder die Größe der Fonts aus und geben den Dateinamen an. Achten Sie darauf, daß alle die Pixel, die außerhalb der 8x8 bzw. 8x16Grenze liegen, im Font verloren gehen.

"Clipboard laden"

Es soll ein Font aus dem Clipboard geladen werden. Dazu muß zum einen ein Font im Klemmbrett vorhanden sein und zum anderen überhaupt ein Klemmbrett existieren. Danach wird wie bei "Öffnen..." verfahren.

"Clipboard speichern"

Ein Font soll ins Klemmbrett gespeichert werden. Auch hierbei muß der Zugriff auf das Klemmbrett möglich sein. Sonst wird wie bei "Sichern..." verfahren.
Achtung: Es werden alle Dateien, die sich im Klemmbrett befinden, vorher gelöscht!

"Clipboard löschen"

Nach der Bestätigung einer Sicherheitsabfrage werden alle Dateien im Klemmbrett gelöscht.

"Ende" (Control Q)

Hier kann das Programm verlassen werden. Sollte sich noch ein geänderter Font im Speicher befinden, so erscheint eine Sicherheitsabfrage, die ihnen die Möglichkeit zum Abbruch gibt.

Das Menü "Block"

GEMFONT erlaubt das Markieren eines rechteckigen Teils eines bestimmten Zeichens. Dieser Ausschnitt wird "Block" genannt und kann ausgeschnitten, kopiert oder gelöscht werden und auch zwischen Fonts ausgetauscht werden. Es kann jeweils nur ein Block zu einer Zeit existieren.

"Blockanfang" (Control B)

Die Maus verwandelt sich in ein Fadenkreuz, mit dem Sie auf den Punkt im Editorraster klicken können, an dem der Anfangspunkt des Block liegen soll. An diesem Punkt erscheint dann als sichtbares Zeichen ein Diagonalkreuz, sofern er nicht gesetzt ist. Ein Klick außerhalb des Gitters bricht die Funktion ab.

"Blockende" (Control K)

Analog zu "Blockanfang" wählen Sie den Endpunkt des Blocks aus. An allen vier Eckpunkten des Blocks erscheint dann ein Diagonalkreuz, der Block ist vollständig markiert. Auch hier bricht ein Klick außerhalb des Gitters die Funktion ab.

"Alles auswählen" (Control A)

Hier wird ein Block definiert, der die gesamte Editormaske umfaßt. Auf diese Weise können schnell komplette Zeichen bearbeitet werden.

"Ausschneiden" (Control X)

Das durch einen Block definierte Rechteck wird aus dem Zeichen, in der Block definiert wurde, entfernt (d.h. ein Teil des Zeichens wird gelöscht) und in einen internen Puffer kopiert. Diese Funktion kann auch mit den Tasten Backspace und Delete ausgelöst werden.

"Kopieren" (Control C)

Wie "Ausschneiden", doch das Gebiet, aus dem der Block ausgeschnitten wurde, wird nicht gelöscht.

"Einfügen" (Control V)

Der vorher ausgeschnittene oder kopierte Block kann in ein beliebiges Zeichen eines beliebigen Fonts wieder hineinkopiert werden. Dieser Vorgang kann so oft wiederholt werden, bis ein neuer Block definiert wird.

Die Maus verwandelt sich die wieder in ein Fadenkreuz, mit der Sie die linke, obere Ecke angeben, an die der Block kopiert werden soll. Die Art, wie der Block eingefügt wird, hängt von der Einstellung "Blockkopiermodi..." ab.

"Blockkopiermodi..."

Das Einfügen des Blocks kann auf verschiedene Arten geschehen. Der Block kann ersetzend eingesetzt werden oder aber vor dem Einsetzen mit dem Zielgebiet verknüpft werden.

Nach Anwählen dieses Menüpunkts erscheint ein PopupMenü an der aktuellen Mausposition, wo Sie den neuen Blockkopiermodus einstellen können.

Das Menü "Puffer"

Jeder Font besitzt 10 frei verfügbare Puffer. Diese können vorübergehend beliebige Zeichen aufnehmen und wieder an beliebige Fontpositionen kopieren. Puffer können zudem geladen und gespeichert werden. Der Unterschied zum Block besteht darin, daß Puffer nicht zwischen Fonts ausgetauscht werden können (zumindest nicht direkt) und daß nur komplette Zeichen in Puffern abgelegt werden können.

"Maske nach Puffer..." (Alternate P)

Das aktuelle Zeichen wird komplett in einen der 10 Puffer, den sie auswählen können, kopiert.

"Puffer nach Maske..." (Alternate M)

Ein Zeichen aus einem der 10 Puffer, den sie auswählen können,

wird in den aktuellen Zeichenpuffer kopiert. Dabei wird immer überlagert kopiert, die Blockkopiermodi wirken sich nicht auf die Pufferfunktionen aus.

"Puffer speichern..." (Alternate U)

Ein einzelner Puffer kann auch gespeichert werden. Wählen Sie dazu einen Puffer sowie den Dateinamen aus, unter dem er abgespeichert wird. Die Pufferdatei sollte immer die Endung "*.GPU" tragen. In sie wird neben ein paar Verwaltungsinformationen nur die Fontdaten geschrieben.

"Puffer laden..." (Alternate O)

Ein gespeicherter Puffer wird eingeladen. Dabei kann der Puffer auch in andere Fonts außer demjenigen, aus dem er heraus abgespeichert wurde, geladen werden. Sollte das Zeichen im Puffer zu groß sein, so wird nur der passende Teil kopiert. Da man auch dem Dateinamen der Pufferdatei nicht ansieht, was sie beinhaltet, zeigt eine Informationsbox die wichtigsten Daten des Puffers an. Darunter fallen die Zeichenausmaße sowie der Name des Fonts, aus dem er ausgeschnitten wurde. Sagt Ihnen der Puffer zu, so können Sie ihn anschließend laden.

Das Menü "Parameter..."

"Speicherstatistik..."

Hier können Sie sich einen Überblick über den aktuellen Speicherverbrauch verschaffen. Dabei werden alle sieben Fonts durch ihren Dateinamen angezeigt. Dahinter wird angezeigt, wieviel Speicher der Font im Editor verbraucht, diese Anzeige ist keineswegs identisch mit der Größe, die der Font einmal in gespeicherter Form einnehmen wird! Das Kästchen vor dem Dateinamen ist durchkreuzt, wenn der Font inzwischen verändert wurde. Ist ein Fontplatz im Speicher nicht belegt, dann wird die dazugehörige Zeile grau dargestellt. Der zweite Teil der Dialogbox zeigt den momentan freien Betriebssystemspeicher an.

"Parameter einstellen..." (Alternate Q)

GEMFONT erlaubt das Einstellen zahlreicher Parameter:

"Grow/Shrink/MoveBoxes anzeigen"

Die Geisterrahmen, die das Öffnen, Verschieben und Schließen von Fenstern sowie von Dialogen begleiten, können hier abgeschaltet werden. Sie sehen zwar effektiv aus, halten aber erfahrene Anwender meist zulange auf.

"Immer *.FNT als Endung voraussetzen"

GEMFONT kann angewiesen werden, statt der Endung "*.FNT" die Endung "*.*" in der Dateiauswahlbox vorauszusetzen. Eine Veränderung dieser Einstellung ist meist unnötig.

"Keine Zusatzmeldungen ausgeben"

Einige Aktionen werden von zusätzlichen Hinweisen zur Bedienung oder Warnungen begleitet (z.B. alle Aktionen, die die Maus in ein Fadenkreuz zur Auswahl verwandeln). Diese können jedoch lästig werden und hier abgeschaltet werden.

"Klingelzeichen bei Alertbox"

Als zusätzliches akustisches Signal kann jede AlertBox von einem "Ping" begleitet werden. Beim Abschluß längerdauernden Aktionen wird ebenfalls ein "Ping" ausgegeben.

"Vsync() bei der Ausgabe benutzen"

Bei schnellen GEMAusgaben kann es zu optischen Unregelmäßigkeiten wie Flackern kommen. Durch Einschalten der VsyncOption kann dies weitestgehend unterbunden werden. Sollte es Probleme mit Grafikkarten geben, dann schalten Sie diese Option ab.

"Fenster dürfen negativen Rand benutzen"

Fenster können zusätzlichen Arbeitsplatz gewinnen, wenn sie ihren linken Rand im negativen Koordinatenbereich ablegen (beispielsweise TEMPUS). Dieses Feature funktioniert zwar, ist jedoch nicht dokumentiert. Soll es unterbunden werden, so ist dieser Schalter zu aktivieren.

"Langsame Dialoge auslagern"

Das Unterbringen von Dialogen in Fensters ermöglicht zwar den Mehrprogrammbetrieb, bremst aber dafür den Aufbau und das Redraw der Dialoge merklich ab, da ein komplizierteres Verfahren zum Erneuern der Darstellung nötig ist (u.a. muß berücksichtigt

werden, ob Teile des Dialogs von anderen Fenstern verdeckt werden.) Wer in einer Einprogrammumgebung arbeitet oder es verschmerzen kann, wenn GEMFONT einmal für einige Sekunden andere Prozesse komplett blockiert, der kann GEMFONT anweisen, besonders zeitintensive Dialoge normal, also nicht in Fenstern darzustellen. In diesem Fall kann die normale und sehr schnelle Redrawroutine verwendet werden.

"Vorgabe 'L30' statt 'P24' bei Import"

Wer vorrangig SignumLaserdruckerfonts importiert, kann einstellen, das statt der Endung "*.P24" (360 DPI) die Endung "*.L30" (300 DPI) zur neuen Vorgabe werden soll.

"Zusätzliche Zeilenvorschübe"

Bei der Testausgabe auf einem Drucker führen manche Druckertreiber einen oder gar mehrere Seitenvorschübe aus (beispielsweise mein NECP.SYSTreiber aus dem FontGDOSPaket, der gleich zwei Seitenvorschübe veranläßt). Um das Papier jedoch platzsparend zu nutzen, spannen Sie in diesem Fall das Papier wieder an die Anfangsposition ein. GEMFONT führt dann selbständig sovielen Zeilenvorschübe aus, daß unterhalb der letzten Position weitergedruckt wird. Führt Ihr Druckertreiber jedoch keinen Seitenvorschub aus, dann deaktivieren Sie diesen Schalter.

"DruckerReset vor Ausgabe ausführen"

Manche Druckertreiber stellen die von ihnen verstellten Druckerparameter nicht mehr zurück (dazu zählt ebenfalls der NECP.SYSTreiber aus dem FontGDOSPaket). In einem solchen Fall kann GEMFONT vor jeder Druckerausgabe einen Reset auslösen. Dazu wird die Kombination "ESC @" verschickt, die auf allen Druckern den gewünschten Effekt haben sollte.

"Druckernummer"

Die Druckernummer, unter der GDOS den Drucker ansprechen soll, kann mit einem Slider zwischen 21 und 30 eingestellt werden. So können mehrere Druckertreiber gleichzeitig installiert und benutzt werden.

"Klemmbrettüberwachung"

GEMFONT kontrolliert in regelmäßigen Abständen, ob eine andere Applikation einen Font im Klemmbrett abgelegt hat. In welchen Intervallen dies erfolgen soll, kann hier eingestellt werden.

Beim Diskettenbetrieb sollte die Überprüfung komplett abgestellt werden.

"Parameter speichern..."

Die Parameter können unter dem Namen "GEMFONT.INF" an dem Ort abgespeichert werden, von dem aus auch GEMFONT gestartet wurde. Findet GEMFONT diesen Ort nicht, so müssen Sie ihn selbst vorgeben. Geben Sie als Dateinamen immer "GEMFONT.INF" an. (Für Fortgeschrittene: GEMFONT sucht seine Infodatei mittels shel_find.)

Das Menü "Fonts"

"Öffne xxxxxxxx.xxx" (Control 1 bis 7)

Wenn ein Fenster geschlossen wurde, so kann es mit diesem Menüpunkten wieder geöffnet werden. Die x's werden dabei durch den Dateinamen des Fonts ersetzt. Beachten Sie dabei, daß pro Font nur ein Fenster geöffnet werden kann.

Das Editorfenstermenü

Die hier besprochenen Funktionen sind in ihrer Wirkung ausschließlich auf den Font, zu dem dieses Editormenü gehört, beschränkt. Einige der Funktionen verlangen explizite Koordinatenangaben. In diesem Fall wird die Maus in ein Zielkreuz verwandelt, mit dem Sie dann das Pixel aussuchen, das die gewünschten Koordinaten besitzt. Ein Klick außerhalb des Editorrasters wird von GEMFONT als Aufforderung zum Abbruch der Funktion interpretiert. Bei der Beschreibung wird nicht mehr jedesmal auf diesen Umstand hingewiesen.

Auch hier gilt, daß nahezu alle Dialogboxen eine Abbruchmöglichkeit besitzen. In diesem Fall werden die von Ihnen gemachten Änderungen innerhalb des Dialogs nicht übernommen. Funktionen, die sich auf den ganzen Zeichensatz auswirken, beeinflussen nur alle diejenigen Zeichen, die zwischen **first_ade** und **last_ade** liegen.

Für viele Funktionen gilt: die Beschreibung klingt komplizierter als die Wirklichkeit. Deshalb sollten Sie mit allen Funktionen ein wenig spielen, um ihre Funktionsweise zu verstehen.

Das Menü "GEMFONT"

"Über GEMFONT..."

Dieser Menüpunkt ist identisch mit dem des Hauptmenüs.

Das Menü "Zeichen"

"Auswahl..." (Funktionstaste 1)

Hier können Sie das Zeichen auswählen, das Sie bearbeiten wollen. GEMFONT bietet Ihnen drei verschiedene Möglichkeiten, ein bestimmtes Zeichen auszuwählen:

Mit dem Slider können Sie alle erlaubten Zeichencodes anwählen. Dabei wird auf die **first_ade** und **last_ade** geachtet.

Sie können das Zeichen direkt per Tastatur eingeben. Damit sind jedoch nicht alle Zeichencodes erreichbar.

Mit "Vorheriges" bzw. "Nachfolgendes" können Sie das Zeichen, das unmittelbar vor bzw. nach dem aktuellen Zeichen liegt, auswählen.

Bei der Auswahl ist folgendes zu beachten: Wenn ein Zeichen direkt eingeben wird, so wird dessen Code als neues aktuelles

Zeichen betrachtet (außer bei "Abbruch"). Die direkte Zeicheneingabe hat dabei Vorrang vor der Sliderauswahl. Wurde ein Zeichen direkt eingegeben und die Button "Vorheriges" oder "Nachfolgendes" aktiviert, dann haben sie keine Wirkung auf den Code des *direkt eingegebenen Zeichens*. Ist dieses Feld leer, dann bezieht sich die Wirkung der Buttons auf das bisherige aktuelle Zeichen (Wenn Sie es erst mal verstanden haben, werden Sie die Effektivität dieser Auswahl feststellen.)

"Breite" (Funktionstaste 2)

Hier stellen Sie mit dem Fadenkreuz die neue Breite des Zeichens ein. Klicken Sie dabei auf diejenige Spalte, die als letzte benutzte Spalte gelten soll. Diese Spalte und die links von dieser liegenden gehören dann zum benutzten Teil des Zeichen. Da mit dieser Methode die Auswahl der Zeichenbreite 0 nicht möglich ist, muß in diesem Fall zusätzlich beim Mausklick eine der ShiftTasten gedrückt werden. In diesem Fall wird die Breite auf 0 gesetzt, egal in welche Spalte Sie geklickt haben.

"Hor. Verschiebung..." (Alternate V)

Das Einstellen des horizontalen Offsets eines Zeichens geschieht hier. Standardmäßig besitzt jedes Zeichen die Verschiebung 0. Insgesamt sind nur die Werte 0 bis einschließlich 255 erlaubt.

"Zentrieren..." (Funktionstaste 7)

Das aktuelle Zeichen wird zwischen dem linken Rand und derjenigen Spalte, die Sie mit der Funktion "Breite" eingestellt haben, horizontal zentriert. Diese Funktion ist wichtig bei Ziffern, die alle die gleiche Breite besitzen müssen, da sonst ein Tabellensatz unmöglich ist. (Vielen SignumFonts ist dies trotzdem neu.)

"Ausmaß anpassen..." (Funktionstaste 8)

Hier können die Ränder aller Zeichen neu eingestellt werden. GEMFONT kann für Sie automatisch die Zeichenober und unterkanten ermitteln. Dazu wird einfach die letzte oberste und unterste benutzte Pixelzeile gesucht. (Praktisch bei importierten Fonts.) Wenn Sie einen Wert in die editierbaren Felder eingeben, dann positioniert GEMFONT alle Zeichen so, daß sie die von Ihnen

vorgegebene Anzahl von freien Pixelspalten auf der rechten/linken Seite besitzen. Diese Funktionen werden dann aktiviert, wenn Sie einen Wert in ein Feld eintragen. Soll die Funktion nicht ausgeführt werden, dann muß dieses Feld leer sein. Die von Ihnen eingestellte Breite der einzelnen Zeichen wird automatisch angepaßt.

Die Funktion "Ziffernbreiten anpassen" ermittelt die größte Breite aller Ziffern, paßt die Breite aller Ziffern an den gefundenen Maximalwert an und zentriert anschließend alle Ziffern horizontal.

Die Anpassung der Breite ist besonders bei vielen SignumFonts sehr hilfreich, eine anschließende Überprüfung und Korrektur von Hand ist dennoch nötig.

"Werkzeuge..." (Funktionstaste 3)

An dieser Stelle finden Sie in einem RaveUpMenü, das an der aktuellen Mausposition erscheint, eine Auswahl von nützlichen Zeichenfunktionen.

"Linie"

Nach der Angabe von zwei Koordinatenpunkten wird eine durchgehende Linie gezogen.

"Kreis"

Als erstes geben Sie den Mittelpunkt des Kreises an, der zweite von Ihnen zu setzende Punkt gibt den Radius an, der Rand wird durch diesen Punkt verlaufen.

Genaugenommen wird mit dem zweiten Punkt nur der Radius des Kreis(segmentes) bestimmt, so daß seine Richtung vom Mittelpunkt aus betrachtet egal ist.

"Rechteck"

Nach der Angabe der Koordinaten von zwei diagonal gegenüberliegenden Punkten eines Rechtecks wird ein ausgefülltes Rechteck gezeichnet.

"Rand"

Wie "Rechteck", aber es wird nur der Rand gezeichnet.

"Füllen"

Geben Sie einen Punkt innerhalb einer Umrandung an, die gefüllt

werden soll.

"Umrandung"

Von dem Zeichen wird nur ein es umgebender Rand gezeichnet. Die Ausmaße des Zeichens wachsen in alle Richtungen um ein Pixel.

"Aushöhlen"

Wie "Umrandung", doch von dem Zeichen bleibt nur der Rand stehen. Die Ausmaße verändern sich nicht.

"1. Quadrant" bis "4. Quadrant"

Wie "Kreis", aber es wird jeweils nur ein Kreisviertel in dem gewählten Quadrant gezeichnet.

"Bogen oben/unten/rechts/links"

Wie "Kreis", aber es wird in der angegebenen Richtung vom Kreismittelpunkt aus ein Halbkreis gezeichnet.

"Schieben/Rotieren..." (Funktionstaste 4)

Zeichen können um einen beliebigen Wert innerhalb ihrer Maske verschoben bzw. rotiert werden. Geben Sie dazu die Verschiebe/Rotierrichtung sowie die Anzahl der Pixel an, um die verschoben/rotiert werden soll. Ein leeres Wertfeld wird mit dem Wert 1 gleichgesetzt. Beim Schieben gehen Pixel, die über den Rand geschoben werden verloren, beim Rotieren erscheinen sie hingegen wieder auf der gegenüberliegenden Seite.

"Spiegeln/Drehen..." (Funktionstaste 5)

Zeichen können an ihrer X oder YAchse gespiegelt werden. Zudem können Sie um 90, 180 und 270 Grad in beide Richtungen gedreht werden. Bei Drehungen um 90 bzw. 270 Grad können Pixel verloren gehen, wenn die Breite des Rasters ungleich dessen Höhe ist. Die Drehung um 180 Grad ist ein Synonym für die Spiegelung an der XAchse.

"Stauchen/Strecken..." (Funktionstaste 6)

Ein Zeichen kann sowohl vergrößert als auch verkleinert werden.

Geben Sie dabei an, ob in horizontaler, vertikaler oder in beiden Richtungen gestreckt/gestaucht werden soll. Der von Ihnen eingegebene Wert gibt das Verhältnis der Größenänderung in Prozent an (Beispiel: Strecken um den Faktor 2: $2 * 100 = 200$, Stauchen um den Faktor 4: $(1 / 4) * 100 = 0.25 * 100 = 25$). Es können auch alle Zeichen um die eingestellten Parameter vergrößert/verkleinert werden.

Seit Version 1.10 werden die Vergrößerung und Verkleinerung von zwei getrennten Algorithmen behandelt, die jeder für sich neu entwickelt und extrem optimiert wurde. Im Gegensatz zum alten Algorithmus konnte die Anzahl der Fließkommaoperationen auf ein Minimum gedrückt werden, innerhalb der notwendigen Schleifen tauchen nur noch elementare Integeroperationen auf, was auch der Grund für die Geschwindigkeitssteigerung um den Faktor 10 bis 15 ist.

Der anzugebende Schwellenfaktor wirkt sich nur bei Verkleinerungen aus: es wird errechnet, wieviele Pixel aus dem Quellraster nach einer Verkleinerung auf den gleichen Punkt im Zielraster fallen, die betroffenen Pixel aus dem Quellraster werden zu einem logischen Bildblock zusammengefaßt. Anschließend wird das Verhältnis der tatsächlich gesetzten Punkte in einem solchen Bildblock zu der Anzahl der gesamten Punkte darin berechnet, der sich ergebende Koeffizient wird mit dem Schwellenwert verglichen, wird er überschritten, dann wird der Punkt im Zielraster gesetzt, sonst nicht.

Ein Schwellenwert von 50 Prozent ist in den meisten Fällen ausreichend, doch bei starker Verkleinerung dünne Zeichen zu stark aus, der Schwellenwert ist hier zu hoch. In solchen Fällen sollte man ihn auf 40 oder gar 30 Prozent absenken. Bei Vergrößerungen stellt sich dieses Problem nicht, da in keinem Fall zwei Punkte aus dem Quellraster auf einen gleichen Punkt im Zielraster fallen können.

Die Aktion kann durch Drücken einer ShiftTaste abgebrochen werden, die bereits veränderten Zeichen werden **nicht** in ihren Ursprungszustand zurückversetzt.

"Verfetten"

Hier kann das aktuelle Zeichen horizontal oder vertikal um den Wert 1 verfettet werden. Der Effekt ist der gleiche wie bei **thicken**.

"Komplett löschen" (Alternate L)

Die Zeichenmaske wird nach Bestätigung einer Sicherheitsabfrage

komplett gelöscht.

"Zeichen aufräumen"

Bei allen Zeichen werden diejenigen Pixel, die außerhalb der Zeichenzellenunterkante bzw. der Zeichenbreite liegen, gelöscht. Zeichen, in denen kein Pixel gesetzt ist, wird die Breite 0 zugewiesen (Ausnahme: das Leerzeichen (Code 32)).

Das Menü "Editor"

"Editormaskengröße..." (Alternate C)

Die von Ihnen gewählte Größe der Editorpuffer kann hier nachträglich geändert werden. Dazu muß aber noch mindestens ein Fontplatz unbenutzt sein. Bei Verkleinerung der Größe gehen die überstehenden Pixel ohne Warnung verloren.

"Editorrastergröße..." (Alternate R)

Die Größe des Editorrasters kann getrennt für die Spalten und Zeilen eingestellt werden. Vorgabe sind dabei die Breite des Systemzeichensatzes. Kleine Werte verschaffen einen guten Überblick, bremsen aber den Bildschirmaufbau.

"Hilfspunkt setzen" (Funktionstaste 9)

Zur besseren Orientierung beim Entwurf von Zeichen können bis zu 32 Hilfspunkte an frei wählbaren Punkten des Fonts gesetzt werden. Diese erscheinen in grauer Rasterung, haben aber sonst keine Auswirkung auf den Font. Zum Setzen eines Hilfspunkt wählen Sie einfach den gewünschten Punkt aus.

"Hilfspunkt löschen" (Alternate W)

Wollen Sie einen Hilfspunkt wieder entfernen, dann können Sie die mit dieser Funktion tun. Klicken Sie den zu löschenden Hilfspunkt einfach ein.

"Alle Punkte löschen"

Nach Bestätigung einer Sicherheitsabfrage werden alle Hilfspunkte gelöscht.

"Änderungen zurücknehmen" (UNDO)

Wenn Sie ein neues Zeichen auswählen, dann wird dieses Zeichen zusätzlich in einen internen Puffer kopiert, bevor Sie es verändern können. Wenn Sie Veränderungen gemacht haben, diese aber wieder rückgängig machen wollen, dann können Sie dies mit dieser Funktion tun. Wenn Sie ein anderes Zeichen auswählen, dann werden alle Veränderungen am alten Zeichen übernommen, eine Rücknahme ist nicht mehr möglich.

"Undopuffer aktualisieren" (Control UNDO)

Diese Funktion finden Sie nicht im Menü, da sie nur selten gebraucht wird. Das aktuelle Zeichen wird mit dieser Funktion zu (fast) jedem beliebigem Zeitpunkt in den internen Puffer kopiert. Ein Druck auf Undo holt dann wieder genau diese Kopie wieder zurück, sofern Sie nicht inzwischen ein anderes Zeichen ausgewählt haben. Das manuelle Erneuern kann bei bestimmten schwerwiegenden Veränderungen als Sicherung nötig sein.

Das Menü "Fontheader"

"Zeichenextrema..." (Alternate Z)

Hier stellen Sie die Parameter **first_ade** und **last_ade** mittels zweier Slider ein.

"Fontlinien..." (Alternate A)

Die Ausrichtungslinien werden hier eingestellt. In einem RaveUpMenü wählen Sie diejenige Linie aus, die sie neu setzen möchten. Anschließend wählen sie Zeile aus, an die die Linie gesetzt werden soll. GEMFONT achtet darauf, daß die Linien in der korrekten Reihenfolge vorliegen und warnt Sie gegebenenfalls. (Ein beliebter Fehler bei alten GDOSFonts!)

"Fontflags..." (Alternate B)

Die Bits 0 bis 3 des Parameters **flags** können hier gesetzt bzw.

gelöscht werden. Deren Bedeutung wurde bereits erklärt.

"Fontkenndaten..." (Alternate D)

Die Parameter **name**, **id** und **points** werden hier eingegeben. GEMFONT kann anhand der Ausrichtungslinien selbst den Wert **points** berechnen, doch dabei orientiert das Programm sich an der momentanen Bildschirmauflösung, weshalb diese Funktion nur für Bildschirmfonts geeignet ist.

"Kursivschrift..." (Alternate K)

Der Wert **skew** wird hier bitweise eingestellt. **left_offset** und **right_offset** werden durch den Button "Größe" automatisch errechnet. Dies geschieht auch, wenn der Dialog mit "OK" verlassen wurde. Die Buchstaben und Zahlen über den Kästchen stehen für das dazugehörige Bit (ausgedrückt als Hexadezimalzahl).

"Fettschrift..." (Alternate F)

Hier wird der Wert **thicken** eingestellt.

"Unterstreichen..." (Alternate G)

Hier wird der Wert **ul_size** eingestellt.

"Helle Schrift..." (Alternate H)

Der Wert **lighten** wird hier bitweise eingestellt.

Schriftproben

Mit der Taste ESC erreichen Sie einen Dialog, mit dem Sie Schriftproben auf dem Bildschirm und dem Drucker erstellen können. Sie können entweder nur das aktuelle Zeichen oder eine von Ihnen vorgegebene Zeichenkette ausgeben lassen. Wenn kein Drucker verfügbar ist, dann ist als Ausgabegerät nur der Bildschirm möglich. Wurden bei der Druckerausgabe zusätzliche Zeilenvorschübe gewünscht, so wird nach jedem Ausdruck der Vorschubszahlen automatisch um den Wert 4 erhöht. Sie können

aber auch eigene Werte vorgeben, beispielsweise 0 beim Beginn einer neuen Seite.

Seit Version 1.10 ist es endlich möglich, auch die VDITextattribute in die Schriftprobe mit eingehen zu lassen. Derzeit können die Attribute "unterstrichen", "kursiv", "fett" und "hell" beliebig miteinander kombiniert und in ihrer Wirkung auf die Schrift betrachtet werden. Das Attribut "outlined" wird nicht unterstützt, weil es kaum gebraucht wird und ich zur Zeit nicht die Notwendigkeit für die (doch recht komplizierte) Implementierung sehe.

Häufige Fragen, Probleme und Tips

viele alte GDOSse (etwa die Versionen 1.0, 1.1 und 1.8 von Atari) haben Probleme, Fonts im MotorolaFormat zu verarbeiten. Berichten zufolge sollen die Reaktionen von leicht verärgert über Pixelmüll bis hin zur Bombenstimmung reichen. Da ich nicht selbst solche Uraltversionen besitze, muß ich mich hier auf Hörensagen stützen. Sollten Sie selbst noch mit solchen veralteten GDOSsen arbeiten, so sollten Sie entweder auf Fonts im MotorolaFormat verzichten oder aber schleunigst auf modernere Versionen umsteigen (beispielsweise AMCGDOS oder NVDI).

Sollten Sie noch mit dem original AtariGDOS arbeiten (diejenigen Versionen vor AMCGDOS), so rate ich Ihnen *dringend*, auf andere GDOSse umzusteigen. Dafür sprechen folgende Gründe:

Die AtariGDOSse sind stark fehlerbehaftet, vieles funktioniert nicht so, wie es sein sollte, und wenn, dann meist nur teilweise.

Die AtariGDOSse bremsen alle Ausgaben über GEM um ca. 30 Prozent ab (kein Scherz). Dies ist ein unzumutbarer Zustand.

Viele Gerätetreiber arbeiten fehlerhaft oder erschreckend langsam unter AtariGDOS.

Neuere GDOSVersionen besitzen viele inzwischen weit genutzte Erweiterungen.

Sollten Sie noch mit einer solchen Version arbeiten und es gibt Probleme mit GEMFONT oder dessen Fonts, so sollten Sie erst einmal ein anderes GDOS einsetzen und schauen, ob die Probleme noch existent sind, bevor Sie sich an mich wenden.

Die Installation von GDOS selbst ist besonders für Neulinge oft ein echtes Rätsel. Aus verständlichen Gründen verweise ich hier auf die Anleitungen zu AMCGDOS oder NVDI etwa, die sich diesem Problem ausführlich annehmen.

Häufig besteht die Versuchung, einen Font, der in einer bestimmten Auflösung fehlt, durch Vergrößerung oder Verkleinerung von bereits bestehenden Fonts zu erzeugen. Dies ist durchaus möglich, doch sollten Sie sich vor Augen halten, daß die neu entstandenen Fonts intensiv nachbearbeitet werden müssen. Trotzdem ist das Ergebnis meist nicht unbedingt ästhetisch (beispielsweise das Verhältnis Breite zu Höhe), denn das Auge nimmt beispielsweise kleine Fonts anders wahr als große Fonts. Also sollte man sich vorher überlegen, ob es nicht sinnvoll wäre, den fehlenden Font komplett neu zu erstellen, auch wenn dies viel Arbeit ist, doch das Ergebnis ist dann meist

wesentlich besser.

Bei Schriftproben auf 24Nadeldruckern muß die lightenMaske sehr "löchrig" gewählt werden, da die Standardmaske, die immerhin jedes zweite Pixel unterschlägt, ihre Wirkung erst unter einer Lupe entfaltet. Generell kann gesagt, daß bei mageren Fonts die Maske "dichter" sein muß als bei fetten Fonts, da sonst die Serifen in ihrer Feinheit zu stark Schaden nehmen.

Bezugsquellen für GDOS

GDOS

Inzwischen ist GDOS recht weit verbreitet und einfach zu erhalten. AMCGDOS ist bis einschließlich Version 3.21 Public Domain und unter anderem bei Atari Deutschland gegen Einsendung eines frankierten Rückumschlag inklusive Disketten erhältlich (näheres siehe unten).

Alle späteren Versionen von AMCGDOS werden in Zusammenhang mit kommerziellen Produkten (Version 5 beispielsweise mit XAct) oder Sharewareprodukten (eine eingeschränkte Version 4 beispielsweise mit Gemini) vertrieben.

Daneben ist das NVDI in der Version 2.1x wärmstens zu empfehlen, daß neben einer deutlichen Bildschirmbeschleunigung auch ein sehr gutes und erweitertes GDOS mit sich bringt. NVDI wird von der Firma "Bela Computer, Eschborn" vertrieben. Von dieser Firma gibt es auch eine Sammlung von einem Druckerfont in den Auflösungen 120, 180, 240, 300 und 360 DPI sowie einem Dutzend teilweise gepatchter GDOSDruckertreiber, über die ich jedoch nichts genaueres zu berichten weiß.

Von Atari USA gibt es ein neues GDOS mit dem Namen FontGDOS. Wesentliche Neuerungen sind die Fähigkeit, Bezierkurven ausgeben zu können sowie das Fontcaching, d.h. Fonts werden nur noch bei Bedarf geladen, was viel Speicher spart. Zudem liegen diesem Paket eine Fülle von z.T. stark optimierten GDOSDruckertreibern inkl. Konfigurationsprogrammen bei; allein dies macht das Paket schon sehr wertvoll.

Leider bietet FontGDOS auch weniger schöne Seiten; etwa daß es das System um ca. 25 Prozent bremst (so gemessen mit meinem TOS 2.05) und Probleme mit der Speicherverwaltung sowie einigen nachweislich sauber programmierten GEMProgrammen hat (beispielsweise ein krachender Absturz beim Polygontest in GEM_TEST.PRG, weil FontGDOS offensichtlich davon ausgeht, daß standardmäßig Bezierkurven statt Polygonen verwendet werden sollen). Im NewsNet habe ich inzwischen Mitteilungen von anderen Anwendern gelesen, die von weiteren z.T. größeren Fehlern berichten.

Seit Ende April 1993 ist nun endlich SpeedoGDOS für ca. 100, DM erhältlich. Da ich bisher damit keine eigenen Erfahrungen sammeln konnte, muß ich mich hier auf die Fachpresse berufen, die sich jedoch schon im Vorfeld beeindruckt von der Anzahl und der Hochwertigkeit der Schriften zeigt.

GDOSFonts

Bezugsadressen für GDOSFonts anzugeben, fällt schwer. Einige Firmen legten ihren Produkten einige Fonts bei (beispielsweise EasyDraw), doch inzwischen gehen viele Firmen dazu über, nur noch eine absolute Mindestanzahl an Fonts (wenn überhaupt) mitzuliefern. In einigen Mailboxen und FTPServern gibt es Archive, die einen Großteil der verfügbaren Fonts erhalten. Leider sind diese Sammlungen nicht vollständig, meist fehlen die Druckerfonts, oder es gibt keine Fonts für die mittlere Bildschirmauflösung oder Metafonts. Dies liegt jedoch nicht an den Zusammenstellern dieser Sammlungen, sondern einfach an dem verfügbaren Angebot an Fonts.

Meines Wissens bietet lediglich Atari selbst GDOSFonts im Zusammenhang mit dem GDOSKit an, das jeder direkt bei Atari beziehen kann. Um dieses zu erhalten, schicken Sie einen mit 3,00 DM frankierten DIN C5 Umschlag, der wiederum einen an sich selbst adressierten gefütterten DIN C5 Rückumschlag enthält (ebenfalls 3,00 DM), an folgende Adresse:

Atari Deutschland GmbH
Abteilung Kundensupport
Am Kronberger Hang 2
D-6231 Schwalbach

Wichtig: Sie müssen selbst drei Disketten mitschicken (deshalb das hohe Porto) sowie in einem kurzen Brief um die Zusendung des GDOSKits bestehend aus Fonts und Treibern bitten, sonst weiß niemand beim Kundensupport, was Sie wollen. Woraus dieses Kit im einzelnen besteht, kann ich (noch) nicht sagen, da ich erst vor kurzem von dieser Möglichkeit erfahren habe.

Professionelle Druckerfonts für 24Nadler sind mir ansonsten bisher nicht untergekommen, angeblich gibt es eine Firma, die kommerziell GDOSFonts für 300 und 360DPI Drucker vertreibt, bestätigen kann ich dies aber nicht. Lediglich den AtariLaserdruckern liegen meines Wissens einige wenige GDOSLaserfonts bei. (Ich lasse mich gerne eines anderen belehren.)

Wer dennoch auf hochwertige Fonts angewiesen ist, dem bleibt nur SpeedoGDOS übrig. Die Quantität und Qualität der dazugehörigen Fonts ist sehr hoch, doch sollte man sich auch vor Augen halten, das SpeedoFonts kommerziell entwickelt und vertrieben werden. Die Preise für einen Font scheinen sich in der gleichen Preisklasse wie für einen CalamusFont einzupendeln und sind damit wohl nur noch für professionelle und kommerzielle Anwender interessant. Es macht also durchaus Sinn, sich weiterhin auch auf dem Gebiet der Pixelfonts zu betätigen!

Interna

Wie alles anfang

GEMFONT besteht grob gesehen aus zwei Teilen: einem selbst programmierten GEMSkett, und dem Fonteditor selbst, der auf diesem Skelett aufbaut.

Nach meinen ersten größeren GEMProjekten stellte ich schnell fest, daß bei der Programmierung viele wiederkehrende Routinearbeiten anfallen, die sich anbieten, zu eigenen Modulen zusammengefaßt zu werden. So entstanden die ersten Module des Skeletts noch unter MWC 3.0.

Mit der Zeit kamen mehr und mehr Module dazu, um im Herbst 1991 beschloß ich, auf Lattice C 5.5 umzusteigen. Ende Februar 1992 hielt ich den Compiler dann endlich in meinen Händen ("... ja, er wird diese Woche verschickt!").

Erneut begann ich, meine GEMModulbibliothek auf Lattice C zu portieren. Doch schon nach wenigen Wochen war ich über dessen miserables, völlig unbrauchbares deutsches Handbuch sowie über die regelmäßigen Abstürze des Optimierers und des Assemblers dermaßen verärgert (das Geld hätte ich mir sparen können... **!%#@!**), bitte achten Sie auf die Einblendung des Schriftzuges "Kommentar" in der linken oberen Ecke... :), daß ich mich entschied, erneut umzusteigen, diesmal auf den GNU CCompiler V2.3.3 der FSF (siehe auch die Artikelserie im STMagazin ab 8/92). Mit ihm verstand ich mich auf Anhieb bestens, daß Skelett gedieh prächtig, bis es Mitte 1992 nahezu ausgewachsen war. Um es praktisch einzusetzen, beschloß ich, GEMFONT erneut anzugehen, nachdem der erste Versuch vor zwei Jahren aus Zeitmangel gescheitert war. Da ich noch auf die Konzepte aus dieser Zeit zurückgreifen konnte, entwickelte sich GEMFONT recht schnell, die erste komplette Version stand Mitte Oktober 1992. Als letztes Problem war die Sache mit dem Copyright zu klären: nach langem Hin und Her stand fest, das die von GEMFONT benutzten Bibliotheken schon seit langer Zeit von ihren Autoren zu "Public Domain" erklärt worden waren (insbesondere unterliegen Sie *nicht* der "GNU Public Licence for Libraries") und die von GCC erzeugten Kompilate keinen rechtlichen Verbreitungseinschränkungen unterlagen, so daß der Verbreitung von GEMFONT als Shareware nichts mehr im Wege stand (hoffentlich haben die mir alle die Wahrheit erzählt... :))

Ein paar Fakten

Das komplette Programm besteht z.Z aus ca. 18000 Zeilen (ja richtig, GEMFONT ist durch Optimierungen kleiner geworden), die

sich auf 32 CModule, 10 HeaderFiles und 1 kleines AssemblerModul verteilen und insgesamt ungefähr 700 Kbyte Umfang erreicht haben. Sowohl auf das Skelett als auch den Editor entfallen je ca. 9000 Zeilen.

Die Entwicklung fand und findet auf einen Mega STE 4 mit 85 MBFestplatte statt. Als Compiler wird der (meiner Meinung nach sehr gute) GNUCompiler V2.3.3 der Free Software Foundation (FSF) verwendet, den ich jedem empfehlen kann, der genug Speicher hat. Als Libraries wurden die MiNTLibrary PatchLevel 30, die GEMLibrary PatchLevel 27 und die Portable Math Library PatchLevel 21 verwendet.

Dabei wurden alle Dateioperationen nicht direkt über das GEMDOS abgewickelt, sondern der MiNT/GNULibrary überlassen, was sich (so hoffe ich) positiv auf die Zusammenarbeit mit MiNT und MultiTOS auswirken wird. Probleme mit MiNT gibt es jedenfalls nicht (getestet mit MiNT 0.96 PatchLevel 14, MiNT 0.99 PatchLevel 2 und MiNT 1.04).

Laden und Speichern von Fonts

Das GDOSFontformat ist zwar recht speicherplatzsparend, aber nur wenig zum direkten Editieren geeignet, außer man benutzt aufwendige Algorithmen, bei denen ständig große Speichermassen verschoben und kopiert werden müßten. Ich habe aber den Weg der Einfachheit vorgezogen (ja ja, ich weiß...), und deshalb wird der Font bei Laden in einzelne Puffer von konstanter Größe ausgepackt. Dies erfolgt auf die gleiche Weise, wie das VDI seine Zeichen ausgibt: über die Funktion vro_cpyfm. Da in der CharacterOffsetTable alle wichtigen Positionen vermerkt sind, geschieht dies ohne weiteren Rechenaufwand. Beim Speichern wird genau der umgekehrte Weg gegangen: die Puffer werden mittels vro_cpyfm zum "Fontbild" zusammengesetzt.

Darüberhinaus wird vro_cpyfm noch in vielen anderen Funktionen eingesetzt, von den Verfremdungsfunktionen bis zu den Pufferfunktionen hin. Dies bedeutet, daß die Rasterkopierfunktion unbedingt fehlerfrei funktionieren müssen, was heute auch bei Grafikkarten aber der Fall sein dürfte. Kleiner Nach oder auch Vorteil: die Geschwindigkeit dieser Funktion beeinflußt viele andere Funktionen, so kann ein Blitter oder NVDI sie sehr beschleunigen. Eine spezielle Kopieroutine wäre vielleicht schneller ("... das ist dann wieder Arbeit."), dafür wäre der Aufwand aber wieder höher ("... wir Informatiker sind von Natur aus faul." Hallo Michael!)

Fontregistrierung

Die Idee

Hinter der Idee, GDOSFonts "registrieren" zu lassen, steht folgender Gedanke: Da jeder Font eine Kennung (ID) besitzt, ist es notwendig, daß diese ID nicht mehrfach für verschiedene Fonts vergeben wird. Um dies zu verhindern, wird jeder Font registriert, d.h. wichtige Fontdaten werden in eine Liste aufgenommen, die allen zugänglich ist (obwohl eine mehrfache Belegung nicht so schlimm ist wie beim XBRVerfahren). Auf diese Weise kann man auch sehen, welche Fonts bereits existieren, damit man vielleicht das Rad nicht noch einmal erfindet bzw. zeichnet. Wer die XBRRegistrierung kennt, dem wird dieses Konzept bekannt sein.

Die jeweils aktuelle Liste wird mit jedem neuem Update verteilt, kann aber auch bei mir auf Diskette angefordert werden (gleiches Verfahren wie das Anfordern einer Kopie von GEMFONT). Natürlich hängt die Zukunft dieser Idee stark davon ab, wie groß die Resonanz, sprich die Mitarbeit ist (obwohl man das ja eigentlich gar nicht als Arbeit bezeichnen kann). Also ...!

Die Registrierung

Um einen oder mehrere Fonts in die Liste aufnehmen zu können, benötige ich folgende Angaben:

die FontKennung (font_id) (Beispiel: "83")
den FontNamen (Name + Stil) (Beispiel: "Times bold")
für welches Ausgabegerät/Auflösung in welchen Größen verfügbar (Beispiel: "300 DPI: 8,9,10,12 points, High: 4,6,8 points")
Angabe über das Original (Beispiel: "Konvertierte SIGNUMSchrift 'Dutch'", oder "mit GEMFONT erstellt" etc.)
Bemerkungen (z.B. Fehler im Header)
Bei kommerziell vertriebenen Fonts die vollständige Bezugsadresse. Ansonsten geben Sie bitte möglichst genau an, woher man den Font bekommen kann (etwa PDVersand oder bei einem Autor etc.), damit auch andere Benutzer diesen bei Bedarf erhalten können.
Name und Anschrift des Autors des Fonts (sofern bekannt, wenn möglich auch NetzAdresse (Mailbox, FTP etc.))

In jedem Fall benötige ich für Rückfragen unbedingt die Adresse derjenigen Person, der mir Fonts zur Registrierung anbietet. Sollten Sie über InterNet erreichbar sein, dann geben Sie bitte auch ihre Netzadresse an, meine finden Sie weiter oben. Scheuen

Sie sich dabei nicht, auch Fonts registrieren zu lassen, die nicht von Ihnen stammen. Damit die Liste möglichst vollständig wird, erbitte ich auch die Registrierung von Fonts kommerzieller Produkte. Ebenso bitte ich um Ergänzung oder Korrektur von bereits vorhandenen Angaben.

Anfragen

Anfragen bezüglich Fonts sollten Sie zuerst versuchen, anhand einer aktuellen Fontliste abzuklären. Sollten Sie dennoch Fragen haben, so können Sie mit mir Kontakt aufnehmen, entweder per Post oder über das InterNet. Wenn Sie mir schreiben, legen Sie bitte ausreichend Rückporto bei.

Antrag auf Registrierung

Hiermit bitte ich um die Registrierung als Anwender von GEMFONT.
Meine Adresse lautet:

(Vor und Nachname)

(Straße und Hausnummer)

(PLZ und Ortname)

(Land)

Den Sharewarebetrag von DM 30, zahle ich wie folgt:

bar

per Scheck (Verrechnungsschecks bitte auf "Sascha Blank"
ausstellen.)

Folgende Angaben sind freiwillig und sollen dem Programmautor
lediglich helfen, sich besser auf die Ausstattung der Benutzer
einstellen zu können:

Computer (genaue Bezeichnung): _____

Speicherausbau: _____ MByte TOSVersion: _____

Ich besitze folgenden Drucker: _____

Er bietet folgende Auflösung (in DPI): _____

Wie sind Sie auf dieses Programm aufmerksam geworden, woher
haben Sie diese Kopie erhalten?

Ich bin damit einverstanden, daß meine Daten zu
Verwaltungszwecken elektronisch gespeichert werden. Eine
Weitergabe an Dritte findet in aber keinem Fall statt.

(Unterschrift)

(Ort und Datum)