

**Pagefox** ist ein eingetragenes Warenzeichen der Scantronik Mugrauer GmbH, Zorneding, Deutschland

**64Net** ist Copyright von Paul Gardner-Stephen, Morphettsville, Australien

**RAMLink** ist ein eingetragenes Warenzeichen der Firma CMD Inc., USA

Alle Typennamen von Commodoreprodukten (**C64, C128, Amiga, Plus4, REU, VDC, 1541, 1571, 1581, 1351**) sind eingetragene Warenzeichen der Firma ESCOM AG, Heppenheim, Deutschland

**GEOS** ist ein eingetragenes Warenzeichen von GeoWorks, Berkeley, USA

**GIF** ist ein eingetragenes Warenzeichen der Firma CompuServe Inc., USA

**PCX** ist ein eingetragenes Warenzeichen der Zsoft Corporation, USA

**HP Deskjet** ist ein eingetragenes Warenzeichen von Hewlett Packard, Böblingen, Deutschland

**Stylus** ist ein eingetragenes Warenzeichen von Epson, Düsseldorf, Deutschland

**Canon BJC 600** ist ein eingetragenes Warenzeichen von Canon, Krefeld, Deutschland

**GoDot** (bis v 1.23) ist Copyright von Magna Media Verlag AG, Haar, Deutschland

**GoDot** (ab v 1.24) ist Copyright von Arndt Dettke und Wolfgang Kling, Minden, Deutschland

Die Rechte aller weiter erwähnten Produktbezeichnungen liegen bei den zugehörigen Firmen. Aus dem Fehlen der Warenzeichenmarkierung ® bzw. ™ kann nicht geschlossen werden, dass die Bezeichnung ein freier Warenname ist.

© 1995 by Arndt Dettke und Wolfgang Kling.  
Alle Rechte vorbehalten.

---

# Inhalt

<b>INHALT .....</b>	<b>3</b>
<b>BEVOR ES LOSGEHT.....</b>	<b>8</b>
SYSTEMAUSSTATTUNG .....	8
FÜR EINSTEIGER.....	9
<b>REFERENZ .....</b>	<b>13</b>
DER HAUPTBILDSCHIRM.....	13
<i>Image Information</i> .....	14
<i>Screen Controls</i> .....	15
<i>Color Controls</i> .....	16
Palette .....	16
Balancing .....	18
Dithering.....	19
<i>Input/Output</i> .....	20
Lader.....	20
Saver .....	20
<i>Image Operators</i> .....	21
<i>Commands</i> .....	22
DER FILEREQUESTER .....	23
<i>Units</i> .....	23
<i>Fileanzeige</i> .....	24
<i>Tastatureingaben</i> .....	25
<i>Statusanzeige</i> .....	26
<i>Löschen</i> .....	27
UNTERSTÜTZTE DATEIFORMATE .....	27
<i>4Bit GoDot</i> .....	29
<i>Amica Paint</i> .....	30
<i>Basictext</i> .....	30
<i>Blazing Paddles</i> .....	31
<i>Botticelli (Plus4)</i> .....	32
<i>Degas (Atari)</i> .....	33
<i>Diashow</i> .....	33

<i>Doodle</i> .....	34
<i>Flip</i> .....	34
<i>Funpaint II</i> .....	34
<i>GeoPaint</i> .....	35
<i>GIF (MS-DOS)</i> .....	36
<i>Handyscanner</i> .....	36
<i>Hires Bitmap</i> .....	37
<i>Hires Manager</i> .....	37
<i>IFF (Amiga)</i> .....	38
<i>Koala</i> .....	39
<i>Magic Disk EMC</i> .....	39
<i>Mandelbrot Construction Set</i> .....	40
<i>Multicolor Bitmap</i> .....	40
<i>OCP Art Studio (Advanced)</i> .....	41
<i>Pagesetter (Plus 4)</i> .....	41
<i>Paint Magic</i> .....	42
<i>PCX (MS-DOS)</i> .....	42
<i>Printfox/Pagefox</i> .....	43
<i>Printtechnik Digitizer</i> .....	45
<i>Scantronik Digitizer</i> .....	45
<i>Starpainter 128</i> .....	46
<b>BEARBEITUNGSMODULE</b> .....	47
<i>Systemmodule</i> .....	48
. <i>ChangeDir</i> .....	48
. <i>ColorCount</i> .....	48
. <i>FileCopy</i> .....	49
. <i>FileType</i> .....	50
. <i>FirstDrive</i> .....	51
. <i>Histogram</i> .....	51
. <i>HowMany</i> .....	51
. <i>NetCopy</i> .....	52
. <i>NewPointer</i> .....	53
. <i>PatternEd</i> .....	54
. <i>REUTool</i> .....	56
. <i>VDCEd</i> .....	57
. <i>Version</i> .....	57
<i>MineSweeper</i> .....	58
<b>READ ME!</b> .....	58

---

<i>Allgemeine Module</i> .....	59
3D-MosaikMC .....	59
4Bit2PFox .....	59
4Bit2VDC1(2) .....	60
4BitLens .....	60
ApplyColors .....	61
ApplyDither .....	61
ApplyMap(Col) .....	61
AverageColor .....	61
AverageGray .....	62
ClipWorks .....	62
DynaRange .....	63
EDgray(MC) .....	64
Edmono .....	64
Edmono-0.35 .....	65
Edmono-1.8 .....	66
FlickerFixer .....	66
FrameClip .....	66
Gradient .....	67
MCScreenFix .....	67
Mirror .....	67
Negative .....	68
OddSwap .....	68
Quick4(TED) .....	68
Rendered2Raw .....	69
ScrollR8 .....	69
Squeeze2Clip .....	69
StretchClip .....	69
Tile .....	70
TileClip .....	70
Tint .....	70
UpsideDown .....	70
VSmooth(Col) .....	71
<i>Konvolutionsmodule</i> .....	72
Blur High .....	73
Blur Low .....	73
Cartoon .....	73
CleanUp .....	73
CrossHatch .....	74
Convolve .....	74

---

Convolv7Bit .....	75
DeepPress .....	75
Emboss .....	76
Gaussian .....	76
LaPlacian .....	76
MedianFilter .....	76
MedianFilterTED .....	77
MosaikMaxi .....	77
MotionBlur(Col) .....	77
OutlineColor .....	77
OutlineMono .....	77
RandMoveLeft .....	78
RandOblique .....	78
RemIsoPixels .....	78
Sharpen1(3) .....	79
Smear .....	79
Speckle .....	79
WaterColor(.O) .....	79
<i>Druckertreiber</i> .....	80
24Pin.drv .....	80
9Pin.drv .....	80
CanonBJC.drv .....	80
HP.drv .....	81
StylusC.drv .....	82
<b>TUTORIEN</b> .....	<b>83</b>
KOMPONIEREN .....	83
ARBEITEN MIT CLIPS .....	87
<b>ZUM SCHLUSS</b> .....	<b>91</b>
TIPPS UND TRICKS .....	91
DIE INI-DATEIEN .....	95

```
load „godot“,8  
run
```



## Bevor es losgeht

### ***Systemausstattung***

Wir verfolgen seit Beginn des Projekts „GoDot“ die Philosophie, dass alle Fähigkeiten des Programms auch mit der allereinfachsten Ausrüstung – sprich: mit einem C64 und einer 1541 ohne besondere weitere Eingabeinstrumente – zur Verfügung stehen müssen. Diese simple und doch so schwierige Grundbedingung konnten wir zum Vorteil vieler User bis heute einhalten.

Nichtsdestotrotz war uns immer daran gelegen, Hardware, die den C64 leistungsfähiger macht, so umfassend wie irgend möglich auch in GoDot einzubinden. Deswegen können wir auch ein Maximal-C64-System beschreiben, das die tatsächliche Mächtigkeit unseres Programms demonstriert, um nicht zu sagen, zu einem wahren Bedienungs Spaß macht.

Ein solches System basiert auf einem C128 DB (die Blechversion mit dem großen VDC-Videospeicher) mit zusätzlichem 80-Zeichen-Bildschirm. Dazu gehören eine CMD-Festplatte, ein CMD-Ramlink mit eingesteckter Commodore-REU und ein oder zwei 1581-Laufwerke oder vergleichbare Diskdrives. Natürlich wäre als Eingabeinstrument eine Proportionalmaus wie die Commodore 1351 ideal, vielleicht zusammen mit einem guten Joystick. Direkten Zugriff auf Daten einer PC-Festplatte oder eines CD-ROM-Laufwerks kann GoDot mit dem 64Net-Kabel bieten. Außerdem sind ein Scanner und ein Videodigitizer Bildquellen, die man sich zunutze machen sollte. Wer Wert auf große Verarbeitungsgeschwindigkeit legt, mag sich mit einer Flash8 anfreunden – alles funktioniert. Wollen Sie in Spitzenqualität drucken? Schaffen Sie sich einen HP-kompatiblen Laserdrucker an. Oder auch einen farbigen Tintenstrahler. GoDot hilft Ihnen.



## ***Für Einsteiger***

Möglich wär's ja, dass Sie weder Maus noch Joystick besitzen. Vielleicht haben Sie beides aber auch gerade verliehen oder unbrauchbar gespielt, egal. Sie ärgern sich jedenfalls, dass die Cursortasten den Mauspfel nur in Zeitlupe bewegen. Nun, GoDot wäre nicht GoDot, wenn's nicht auch anders ginge.

Die Tasten, die den Pfeil beflügeln, sind „S“, „F“, „H“ und „F3“ zusammen mit der Commodoretaste. Sie täuschen dem C64 vor, mit einem Joystick angesteuert zu werden.

Der Pfeil wird vom Bildschirmrand nicht aufgehalten, da er stört, wenn wir ein Bild betrachten. So können wir ihn nach außerhalb verschwinden lassen und das Bild ist in voller Pracht zu sehen. Probieren Sie's einmal: der Schaltknopf oben rechts der Mitte mit der Aufschrift „Redisp“ ist für die Bildanzeige zuständig. Klicken Sie drauf! (Mit der Returntaste, der linken Maustaste oder dem Feuerknopf am Joystick.) Wenn noch kein Bild geladen wurde, sieht der Bildschirm zwar etwas merkwürdig aus, aber Sie wissen, was gemeint ist. Klicken Sie noch einmal, um die Anzeige wieder zu verlassen. Also: um sich ein fertiges Bild anzuschauen, benutzen Sie *Redisp*.

Weiter rechts haben wir den Namen unseres Programms verewigt, dort finden Sie die Aufforderung „GoDot!“. Nehmen Sie's einmal wörtlich und klicken Sie drauf, es handelt sich ja ebenfalls um einen Schaltknopf, wie alle erhabenen Flächen mit einem dünnen Rand. Erschrecken Sie nicht, sie sehen gerade einen *Guru* (so würden Amigafreaks das nennen). Dabei handelt es sich in GoDot um die (etwas effektvolle) Ausgabe der Versionsnummer des Programms. Wenn Ihre Nummer mit „FF“ aufhört, stammt die Software aus dem Internet, dem weltumspannenden Kommunikationsnetz. Falls Sie selbst angeschlossen sind: eine kluge Investition!

Einige andere Schaltknöpfe führen Sie in einen weiteren Bildschirm, der Ihnen zeigt, was auf Ihrer eingelegten Diskette so zu finden ist. Benutzen Sie einmal einen derjenigen Knöpfe, die in ihrer Aufschrift irgendwie „Load“ oder „Save“ enthalten. Diesen neuen Bildschirm erklären wir später noch genauer (im Kapitel „Der Filerequester“), deshalb hier nur der Hinweis, wie Sie ihn verlassen. Sie betätigen dazu die Schaltfläche „Cancel“ und schon baut sich GoDots Hauptbildschirm wieder auf.

Sie können natürlich (irgendwann später!) auch GoDot verlassen, Sie müssen nur „Exit“ anklicken und sich dann entscheiden, ob Sie wirklich keine Lust mehr haben. Soll's ja geben.

Haben Sie's gemerkt? Die Schaltflächen des Hauptschirms „unter“ dem Exitfenster haben nicht mehr reagiert, nur die innerhalb funktionierten! Das ist eine grundlegende Eigenschaft von GoDot, nur das oberste Fenster ist aktiv. Jedes Fenster enthält daher mindestens einen Knopf, mit dem man es wieder verlassen kann, meist heißt er „Cancel“ oder „Leave“.

Einige Schaltknöpfe ändern ihre Aufschrift beim Auslösen. Es handelt sich um „Einsteller“, sie zeigen immer den momentanen Zustand der zugehörigen Funktion. Wenn Sie unterhalb des „Screen Controls“-Schriftzugs klicken, wechselt die Beschriftung beispielsweise zwischen „Multi“ und „Hires“ hin und her. Die Schaltfläche hinter „Dith:“ bietet noch ein paar mehr Aufschriften, auch die Zahl hinter „Colors“ können Sie auf diese Art Ihren (späteren) Bedürfnissen anpassen.

Ganz unten auf der linken Bildschirmseite finden Sie die zweitwichtigste Schaltfläche im ganzen Programm: „Display“. Aktivieren Sie sie ruhig! Der C64 schaltet jetzt um auf Grafikdarstellung (wie beim „Redis“-Knopf), löscht aber das vorhin gesehene Bild und fängt an,

ein neues Bild aufzubauen. GoDot nimmt hierfür die Bilddaten im sogenannten 4Bitspeicher und rechnet sie um für die Anzeige als wirkliches Bild. Dabei berücksichtigt das Programm allerhand Einstellungen, die Sie selber vornehmen können (Sie haben's ja eben schon probiert), deswegen dauert es ein bisschen bis das Bild fertig aufgebaut ist. Ein weiterer Klick (oder STOP), und die Grafik schaltet sich aus. Wenn Sie keine anderen Einstellungen zwischendurch vornehmen, funktioniert „Display“ jetzt wie „Redisp“. Nach Änderungen wird das Bild wieder neu erzeugt. Also: *Display* berechnet ein neues Bild.

Laden wir doch mal eins! Oben links der Mitte des Bildschirms steht „4BitGoDot“? Gut, dann klicken wir auf die Schaltfläche unter „Redisp“ mit der Aufschrift „Load“. Ein neuer Bildschirm erscheint und füllt sich mit einigen Filenamen. Klicken Sie zweimal kurz hintereinander auf einen Namen, der mit „.4Bit“ endet. GoDot zieht sich jetzt Bilddaten in seinen Speicher. Wie weit es gekommen ist, sehen Sie an der Laufanzeige. Zum Schluss landen Sie wieder im Hauptbildschirm. Wenn Sie nun „Display“ betätigen, gibt's nicht nur Gewusel auf dem Bildschirm, nicht wahr?

Den wichtigsten GoDot-Knopf finden Sie unten rechts. Er trägt die Beschriftung „Execute“. Näheres dazu sparen wir uns erst einmal auf bis zum Kapitel „Image Operators“. Ganz intensiv gehen wir dann in „Bearbeitungsmodule“ darauf ein.

So, das waren die einfachen Tätigkeiten. Ein bisschen kennen Sie sich schon mit GoDot aus, nicht wahr?



## Referenz

### *Der Hauptbildschirm*

Der Hauptschirm ist geteilt in sechs *Fenster* mit unterschiedlichen Ausmaßen und Inhalten. In jedem Fenster befinden sich beschriftete Schaltflächen, die sogenannten *Gadgets*. Ein Gadget wird aktiviert, indem man mit dem Mauszeiger darauf fährt und die Auslösetaste betätigt. Bei einigen ändert sich daraufhin die Beschriftung. Der jeweils sichtbare Text gibt die aktuelle Einstellung an. Man kann durch diese Einstellungen vorwärts oder rückwärts blättern, indem man rechts bzw. links ins Gadget klickt.



Die beiden oberen Fenster tragen keine ausgewiesenen Titel, wir nennen sie „Input/Output“ (linke Seite) und „Commands“ (rechts). Das Gadget *GoDot!* im Commandsfenster zeigt die Versionsnummern der beiden Hauptbestandteile GoDots an („god.main“ und „god.upmem“ auf Diskette).

## Image Information

Beginnen wir mit dem Fenster „Image Information“. Es enthält das größte Gadget auf dem Hauptbildschirm, den *Preview*. Dieses Gadget dient der Vorschau des im Speicher befindlichen Bildes. Es wird in vier Graustufen und geringer Punktauflösung dargestellt, was für eine grobe Übersicht völlig ausreicht. Ein Klick ins Gadget schaltet den Preview an und wieder aus.

Eine weitere Aufgabe des Gadgets besteht darin, dem Anwender zu zeigen, dass ein Modul, das keinen eigenen Requester öffnet, jedoch einige Zeit für seine Arbeit benötigt, noch aktiv ist. In einem solchen Fall wandert ein weißer Strich von oben nach unten über das freie Feld und gibt somit den Fortschritt der Bearbeitung an.



Das Fenster selbst enthält nach dem Start nichts weiter als den Hinweis, dass der Speicher keine bekannten Bilddaten enthält, das Bild heißt zunächst „Untitled“ und weitere Informationen stehen nicht zur Verfügung. Natürlich kann der Bildspeicher trotzdem gefüllt sein, da er während des Bootvorgangs nicht gelöscht wird. Überprüfen Sie das mit dem Preview.

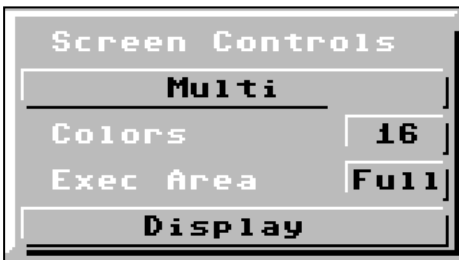
Nach dem Laden eines Bildes erscheinen alle nötigen Bildinformationen neben und über dem Previewgadget: der Name des Bildes, sein ursprüngliches Speicherformat, die Auflösung der 4Bitdaten (320x 200 entspricht in etwa Hires und 160x200 Multicolor) und der

Typ der Bilddaten, entweder „Color“ für Bilder, die an die Farben des C64 angepasst wurden oder „Gray“ (bzw. „Grey“), wenn es sich um Bilder in 16 Graustufen handelt.

## Screen Controls

Die Gadgets dieses Fensters steuern den Rendervorgang eines in Arbeit befindlichen Bildes.

*Exec Area* legt fest, ob das ganze Bild neu aufgebaut werden soll („Full“) oder nur ein vorher mit dem Bearbeitungsmodul *mod.Clip-*



*Works* eingestellter Teilbereich („Clip“).

Die Angabe hinter *Colors* legt fest, in wievielen Grau- oder Farbstufen der eingestellte Bereich gerendert werden soll. Ma-

ximum ist 16 (alle Farben), der unterste Wert ist 2 (monochrom). Der angezeigte Wert wird verändert, indem man links oder rechts ins Gadget klickt. Die Fortschaltung geschieht zyklisch, nach 16 kommt 2 und umgekehrt. Nach einer Änderung sollte man die Palette anpassen, da die vorher gewählten Farben erhalten bleiben.

Das lange Gadget oberhalb von „Colors“ stellt die Grafikauflösung des C64 ein, den *Mode*. Bei „Hires“ rendert GoDot ein Bild mit 320x200 Punkten, „Multi“ ergibt eins von 160x200 Punkten. Je nach Zielformat (in dem das Bild später angezeigt oder gespeichert werden soll), wählen Sie hier den entsprechenden Mode aus.

Wenn Sie ein Bild für den Ausdruck vorbereiten oder für den Export in C64-Hiresformate, bringt die Einstellung „Hires“ die meisten Details. Generell gilt, dass farbige Bilder besser in „Multi“ gerendert werden sollten, monochrome Bilder in „Hires“.

*Display* schließlich startet den Rendervorgang. GoDot wendet dabei alle Einstellungen der Fenster „Screen Controls“ und „Color Controls“ zur Neuberechnung des Bildes an, ohne jedoch die eigentlichen Bilddaten (die 4Bitgrafik) anzutasten. Daher kann eine Bildberechnung jederzeit durch Änderung der Einstellungen zurückgenommen werden. Es eröffnen sich vielfältige Experimentiermöglichkeiten.

## Color Controls

### *Palette*

Das *Palette*-Gadget eröffnet ein neues Fenster, einen sogenannten *Requester*, mit weiteren Gadgets. Nur die Gadgets innerhalb dieses



neuen Requesters können angewählt werden. Die weiterhin sichtbaren Schaltflächen des Hauptbildschirms sind solange nicht mehr aktivierbar wie Sie sich im Palette-

Requester befinden. „Palette“ dient zum Zusammenstellen einer neuen Farbfolge für die weitere Bildberechnung. Die neue Farbpalette wird im unteren Teil des Requesters angezeigt, direkt über den Aktionsgadgets. Ihre Länge passt sich der Angabe hinter „Colors“ im Fenster „Screen Controls“ an.



Sollten Sie einmal versehentlich hier gelandet sein, verlassen Sie den Requester wieder über *Cancel*. Selbst wenn Sie bereits viele Veränderungen mit den übrigen Gadgets vorgenommen hätten, würde nach „Cancel“ alles wieder so sein wie vor dem Aufruf von „Palette“, alles wäre zurückgenommen. Diese Eigenschaft des Cancelgadgets gilt für alle Requester, in denen es auftaucht.

Ähnlich wirkt *Undo* (engl. „rückgängig machen“). Die jeweils zuletzt in diesem Requester vorgenommene Einstellung wird dadurch wieder aufgehoben. Die Aufhebung bezieht sich nicht auf das erneute Rendern mit „Display“.

Das Gadget *Default* stellt die angezeigte Palette so um, dass eine für die jeweilige Farbanzahl (insgesamt 14 verschiedene) optimal zusammengestellte Farbfolge gebildet wird. Diese Farbfolge ist in jedem Falle so strukturiert, dass von links nach rechts immer hellere Farben aufeinanderfolgen.

Sie können Paletten auch von Hand zusammenstellen, indem Sie im *Pick Color*-Gadget die gewünschte Farbe anklicken (worauf sie rechts angezeigt wird) und unten im Palettengadget auf die vorgesehene Position klicken. Die gewählte Farbe ersetzt daraufhin die dort befindliche.

Ein Klick auf *Display* zeigt Ihnen, ob diese Wahl Ihren Vorstellungen entspricht. Das Bild wird neu berechnet wie bei „Display“ in „Screen Controls“.

Wenn Sie zufrieden sind, bestätigen Sie GoDot den jetzigen Zustand der Palette als gültig für die weitere Arbeit, indem Sie *Accept* anklicken.

## ***Balancing***

Auch *Balancing* öffnet einen eigenen Requester mit mehreren Einstell- und Aktions-



gadgets. Das Ganze dient dem Verändern von Helligkeit und Kontrast bei der Neuberechnung der Bild-daten durch „Display“. Den größten Nutzen aus „Balancing“ hat man, wenn

es darum geht, Schwarzweißdrucke zu fertigen oder wenn man 256-Farben-PCX-Bilder einlesen möchte.

Die Pfeile links und rechts von *Brightness* und *Contrast* erhöhen bzw. erniedrigen den jeweils angezeigten Wert, der sich zwischen -16 und +15 bewegen kann. In der Rasteranzeige darunter bekommt man angezeigt, welche der Farben/Graustufen davon betroffen sind. Ist z. B. das Feld ganz links von einem Muster besetzt, wird im Bild kein Schwarz mehr vorkommen.

Mit *Reset* stellt man die beiden Werte auf den mittleren Wert 0 (Null) zurück. Balancingwerte bleiben normalerweise solange bestehen, bis Sie eine neue Einstellung vornehmen. Manche Module restaurieren sie jedoch von sich aus auf die Mittelwerte, wenn es sinnvoll erscheint (z.B. der Lader *ldr.PCXprep4IFLI*).

*Accept* und *Cancel* haben hier wie überall die übliche Bedeutung.

## *Dithering*

Das Kürzel *Dith* steht für „Dithering“, was zu Deutsch etwa „Streuung“ bedeutet. Einstellungen hinter „Dith“ sind immer dann wirksam, wenn es darum geht, monochrome Bilder zu erzeugen. Jede Graustufe (oder Farbe) GoDots bekommt ein einfarbiges Punktraster



zugewiesen, das sie ersetzt, um in rein monochromen Darstellungen (Druck, VDC-Anzeige, Printfox usw.) weiterhin natürlich und abstufungsreich zu wirken.

Beim Start steht die Anzeige auf *Off* und ist somit ausgeschaltet. Alle Helligkeiten bis etwa zur Mitte der Palette werden durch schwarz ersetzt, die zweite, hellere Hälfte bekommt die Farbe Weiß zugewiesen.

Das Punktraster *Ordered* ist gedacht für monochrome Bilder, die hauptsächlich auf dem Bildschirm angezeigt werden sollen. Es vermeidet Punktfolgen, die auf C64-Bildschirmen nicht so dargestellt werden können, wie sie im Speicher vorliegen, z. B. senkrechte Linien mit einer Breite von einem Pixel.

*Pattern* ist ein Raster, das besonders für den Ausdruck geschaffen wurde. Es sieht am Bildschirm grob und lückenhaft aus, auf dem Papier dagegen fein strukturiert. Graustufen werden dort gut angenähert. Das Patternraster kann man mit dem Modul *mod..PatternEd* frei bearbeiten.

Das *Noise*-Dithering versieht die Graustufen und die zugeordneten Farben mit einem Zufallsraster, das der jeweiligen Helligkeit entsprechen soll. Mit diesem Muster gerenderte Bilder sehen jedes Mal anders aus. Am besten gelingen Bilder, wenn der Kontrast verstärkt wurde. Sie wirken daraufhin wie mit Kohle gezeichnet.

Bei *Random* ist ausschließlich die mittlere Graustufe betroffen. Sie wird ebenfalls durch ein Zufallsmuster ersetzt. Ansonsten verfährt GoDot wie beim „Off“-Dither.

## Input/Output

Links oben im Hauptbildschirm haben wir ein kleineres Fenster mit den beiden Aufschriften „Load:“ und „Save:“ (mit Doppelpunkten). Rechts daneben folgt jeweils ein Gadget mit einer wei-



teren Beschriftung. Der Gadgettext sagt Ihnen, welches Lade- bzw. Speichermodul für Grafiken GoDot momentan benutzt.

### *Lader*

Klicken Sie auf das Gadget hinter „Load:“ dann öffnet sich der Filerequester GoDots und zeigt andere Lademodule (*Lader*), sofern die Diskette im angewählten Laufwerk solche enthält. Lader haben auf der Diskette das einleitende Erkennungskürzel „ldr.“ im Namen. Mit Hilfe der Lader haben Sie Zugriff auf die wichtigsten Grafikformate von C64, Amiga und PC.

### *Saver*

Auch das Gadget hinter „Save:“ führt zum Filerequester. Er zeigt diesmal alle Speichermodule (*Saver*) auf der Diskette. Saver sind am

Kürzel „svr.“ zu erkennen. Lader und Saver zusammen bieten Ihnen die einfachste Möglichkeit, ein Grafikformat in ein anderes zu wandeln. Sie können bei der Erarbeitung professionell gestalteter Bilder also immer genau das Grafikwerkzeug zum Einsatz bringen, das Ihnen am geeignetsten erscheint.

## Image Operators

In diesem Fenster werden alle Module installiert und aktiviert, die für die erweiterten Fähigkeiten GoDots zuständig sind, im engeren Sinn alle Bildbearbeitungsmodule (daher die Bezeichnung *Image Operators*), aber im weiteren Sinn auch Module, die den Umgang mit Disketten, zusätzlichem RAM und anderen Ausgabegeräten erleichtern (die sogenannten *Systemmodule*). Solche Module werden installiert wie Saver oder Lader, indem man auf das Gadget hinter *Inst* („Installiert“) klickt. Im Filerequester sieht man daraufhin alle zugehörigen Module. Sie führen das Kürzel „mod.“



Man aktiviert ein installiertes Modul, indem man das *Execute*-Gadget auslöst. Je nach Funktionsumfang des Moduls öffnet sich daraufhin ein neuer Requester mit weiteren Auswahlgadgets, oder es läuft ein feiner weißer Strich von oben nach unten über das Preview-Gadget und zeigt damit an, dass GoDot beschäftigt ist. Während dieser Zeit sind Benutzereingaben nicht möglich. Sie können mit Ihrer Arbeit fortfahren, wenn der Strich verschwunden ist. Eine genaue Beschreibung der Funktionen aller Operatoren finden Sie im Kapitel „Bearbeitungsmodule“.

## Commands

Dieses Fenster enthält eine Reihe von Funktionen, die „Kommandos“ an GoDot darstellen, nicht jedoch dessen Einstellungen verändern. Die Gadgets *Load* und *Save* stehen in unmittelbarem Zusammenhang mit den entsprechenden Schaltflächen im „Input/Output“-Fenster oben links. Der dort aktuell verfügbare Lader bzw. Saver wird über diese beiden Gadgets aktiviert. Fast immer erscheint



darauhin der GoDot-File- requester. Einige Lader für besondere Grafikformate öffnen zunächst einen Vorre- quester zur Auswahl eines bestimmten Modus, manche

Lader starten ohne weitere Anzeige sofort.

*Redisp* stellt das als letztes gerenderte Bild noch einmal dar. Das bleibt so, bis Sie mit „Display“ in „Screen Controls“ diesen Zustand aktualisieren (bis Sie also neu rendern).

*Exit* schließlich führt in eine Sicherheitsabfrage, die verhindern soll, dass Sie unbeabsichtigt GoDot beenden und Ihre Arbeit noch nicht gespeichert war. Erst eine nochmalige Bestätigung Ihres Wunsches, das GoDotsystem zu verlassen, bringt Sie zum eingebauten BASIC des C64. Nach „Exit“ muss GoDot vollständig neu gebootet werden. Es gibt keinen verkürzten Neustart.

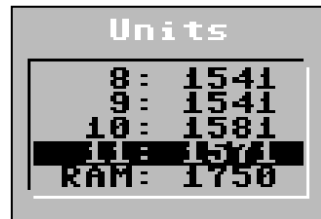
## Der Filerequester

Das zentrale Element GoDots ist der *Filerequester*. Alle anschließbaren Massenspeicher werden hierüber erreicht, seien es Diskettenlaufwerke, Festplatten oder RAM-Erweiterungen.

### Units

Dieses Fenster zeigt durch Unterlegung an, auf welches Laufwerk sich die augenblickliche Lade- oder Speicheraktion bezieht.

Während des Startens hat GoDot die zugehörige Hardware getestet und zeigt hinter der Laufwerksnummer den Typ des Drives (1541, 1571, 1581 usw.) Wenn GoDot nicht herausfinden konnte, was für ein Laufwerk angeschlossen ist, geht es vorsichtshalber von einer 1541 aus.



CMD-Hardware macht GoDot daran kenntlich, dass statt eines Doppelpunktes hinter der Nummer ein „>“-Zeichen erscheint. Handelt es sich zusätzlich noch um eine sogenannte „Native Partition“ (s. CMD-Handbücher), lautet auch der Laufwerkstyp „CMD“.

Wo keine Typenbezeichnung steht, gibt es auch kein Laufwerk.

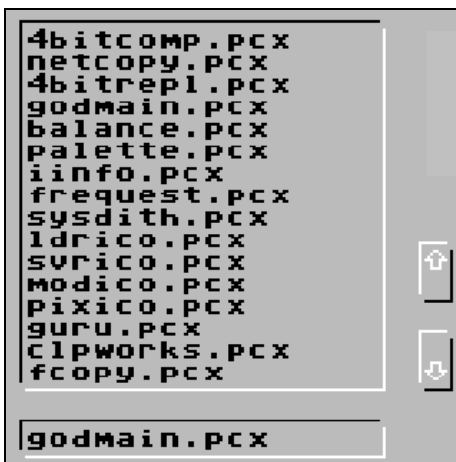
RAM-Erweiterungen werden erkannt und – sofern es sich um REUs handelt – automatisch eingebunden wie ein weiteres Diskettenlaufwerk. Das RAM des C128-Videospeichers („VDC“) stellt GoDot auch zur Verfügung, allerdings nicht so weitgehend wie das der REUs. Man kann dort temporäre Dateien (*svr.Temp*) ablegen, jedenfalls beim „großen“ VDC mit 64KB („VDC2“). Auch das freie RAM eines Pagefox-Moduls kann man zum Zwischenspeichern nutzen (s. aber

auch *mod.4Bit2PFox*). Ist keinerlei zusätzliches RAM erkannt worden, gibt GoDot den Typ des C64 an („C64n“ für den C64 II).

Man wechselt das Laufwerk, indem man auf eine andere Nummer deutet und klickt. GoDot zeigt daraufhin das Inhaltsverzeichnis der eingelegten Diskette.

## Fileanzeige

GoDot listet im großen Fenster links alle Files, die zur angewählten Aktion passen. Will man also einen neuen Lader installieren, erscheinen ausschließlich Lader in der Anzeige, alle anderen Files der Diskette blendet GoDot aus. Diese „Filter“-Einstellung kann man rechts unter dem Cancel-Gadget ablesen, für Lader steht dort das entsprechende Filekürzel



„ldr.\*“. Weitere Kürzel sind „svr.\*“ (Saver), „mod.\*“ (Modifizier), „pat.\*“ (Rastermuster), „ptr.\*“ (Mauszeiger) und einige andere.

Ein einzelner Stern steht für Files, die eventuell Bilder sein könnten.

Klickt man auf einen angezeigten Namen, erscheint er auch im darunterliegenden Eingabegadget. Files, die dort angezeigt werden, kann man mit dem *Load*-Gadget in den C64-Speicher laden. Wenn zweimal kurz hintereinander auf den Namen in der Anzeige geklickt wird (*Doppelklick*), hat das den gleichen Effekt wie „Load“.





Bei gefülltem Anzeigefenster blättert man mit den beiden kleinen Pfeilgadgets die Fileliste auf und ab. War man ganz am Ende, führt ein erneuter Klick auf die Laufwerksnummer im Unitsfenster am schnellsten an den Anfang.

Die User, die mit mehreren Laufwerken arbeiten, werden es sehr begrüßen, dass das jeweils zuletzt angeählte beim nächsten Aufruf automatisch angezeigt wird. Das funktioniert sogar getrennt für Laden und Speichern, so dass ohne Aufwand und schnell gearbeitet werden kann.

Eine REU ist generell vorrangig, da sie am schnellsten ist.

## **Tastatureingaben**

Im *Eingabegadget* unterhalb der Fileanzeige lassen sich auch von Hand Filenamen eintragen. Dazu klickt man einfach in das Gadget hinein. Je nachdem, wo geklickt wurde, setzt GoDot einen nicht blinkenden Textcursor an die betreffende Stelle. Hat man hinter einen Namen geklickt, so steht der Cursor dann hinter dem letzten Buchstaben.

Im Eingabemodus sind Maus und Joystick weiterhin aktiv und können benutzt werden, um den Textcursor an eine neue Position zu setzen (dort klicken). Ansonsten gibt es nun eine Reihe von Tasten und Tastenkombinationen, die das Editieren erleichtern sollen. Mit der folgenden Tabelle haben Sie einen Überblick.

Taste	Funktion
Buchstaben Ziffern	erscheinen wie eingegeben (nicht alle Zeichen stehen zur Verfügung!)
DEL	löscht das Zeichen über dem Cursor (anders als im BASIC-Editor!)
INST	schaltet um auf Überschreibmodus, nochmaliges Drücken führt zurück zum Einfügemodus
HOME	setzt den Cursor an den Anfang der Eingabe
CLR	setzt den Cursor ans Ende (hinter d. letzte Zeichen)
C=X	löscht den Inhalt des Eingabefensters
STOP	bricht die Eingabe ab und nimmt Änderungen zurück
RETURN	beendet die Eingabe und setzt den Mauszeiger auf das „Load“-Gadget. Ein weiteres RETURN startet daraufhin den Ladevorgang

Alle Eingabegadgets GoDots, egal, an welcher anderen Stelle sie auftauchen sollten, werden bedient wie hier im Filerequester.

## Statusanzeige

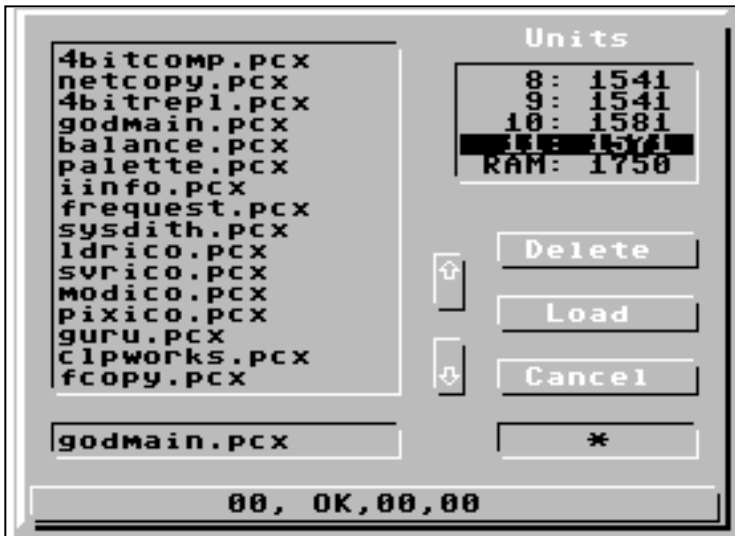
Das lange Gadget unten im Requester ist das *Status*-Gadget. Dort zeigt GoDot die üblichen Floppymeldungen. Wenn einmal (was ungewöhnlich wäre) die Floppy-LED blinken sollte, und GoDot eine Meldung nicht abgeholt hat, klicken Sie einfach das Statusgadget von Hand an (kommt vor bei *mod..FileType*, wenn ein GEOS-Name angeklickt wurde).

Der anderen Funktion des Gadgets begegnen Sie, wenn Sie ein Bild in den Speicher laden. Dann wandert hier ein Laufbalken von links nach rechts durch das Gadget und zeigt so, dass GoDot beschäftigt ist. Manche Lader lassen mehrere Balken nacheinander laufen. Ein kurzer Text erläutert jeweils die gerade stattfindende Aktion.

## Löschen

Das Gadget *Delete* beseitigt das ausgewählte File von der Diskette. Dieser Vorgang lässt sich innerhalb GoDots nicht rückgängig machen, deshalb Vorsicht! Ein einzelner Stern löscht **alle** Dateien des ausgewählten Typs!

Wer unbeabsichtigt eine wichtige Datei gelöscht hat, sollte das System sofort verlassen und das File mit einem geeigneten Utility oder einem Diskmonitor zurückholen. Erst danach kann man beruhigt mit GoDot weiterarbeiten.



## ***Unterstützte Dateiformate***

**G**oDot kennt Lader und Saver für eine ganze Reihe von verbreiteten C64-Grafikformaten. Auch die wichtigsten Formate von Fremdrechnern (PC, Amiga und Atari) haben wir in GoDot eingearbeitet.

Beim Einlesen der Bilddaten wandelt der jeweilige Lader die Informationen grundsätzlich ins goDotinterne 4Bit-Format. Damit steht für ein Abspeichern in einem gegenüber dem Standard-C64 leistungsfähigeren Zielformat (z.B. PCX) immer ein Optimum an Qualität bereit. Manchmal liegen so viele Daten vor, dass sie ein C64 gar nicht adäquat darstellen kann. Ausgegeben als (Farb-)Druck oder in einem Format für einen grafisch mächtigeren Rechner machen aber auch solche Bilder sehr wohl Sinn.

Unter GoDot geladene Bilder können durch Abspeichern mit einem anderen Saver in jedes unterstützte Format konvertiert werden. Es gibt jedoch nicht für alle Formate sowohl einen Lader als auch einen Saver.

Weitere Informationen zu Lade- und Speichermodulen finden Sie in den 64er-Magazinen 9/94 bis 6/95 in der „How to GoDot“-Serie und auf der GoDot-Homepage ([www.godot64.de](http://www.godot64.de)). Programmierer finden dort etwa genauere Angaben zum jeweils verwendeten Packalgorithmus.

Manche Lader starten nicht gleich mit dem Filerequester, sondern lassen zunächst einen Optionsrequester erscheinen, der es erlaubt, den Ladevorgang über bestimmte Voreinstellungen zu steuern. Bei Savern gibt es solche Optionsauswahlen auch, dort werden besondere Einstellungen aber erst **nach** Eingabe des Namens im Filerequester abgefragt.

## 4Bit GoDot

Format: 4Bit, gepackt und ungepackt (126 Diskblocks)

Lader:	ja, 4Bit&Arith 4Bit&Map 4Bit&Mask 4BitGoDot 4BitRGB(d) Morph!64 RawBytes	Saver:	ja, 4BitGoDot Morph!64 Temp
--------	---	--------	--------------------------------------

Anzeige: 4BitGod, 320x200, Gray

Ldr.4BitGoDot wird beim Starten des Systems als Vorgabelader installiert. Kompositionen (ldr.4BitGoDot, ldr.4BitMask) und Bildverknüpfungen (ldr.4Bit&Arith, ldr.4Bit&Map, ldr.4BitRGB) werden ausschließlich durch dieses Format ermöglicht. Im Kapitel „Tutorien“ finden Sie eine Beschreibung zu den wichtigsten genannten 4Bitladern.

Ldr.RawBytes geht davon aus, dass das Bild (oder andere Daten im 4Bit-Format) ohne Startadresse abgespeichert wurde(n).

Einen ausdrücklichen Temp-Lader gibt es nicht, da alle im Namen mit „4Bit“ beginnenden Lader auch das temporäre RAM-File ansteuern können. Zum svr.Temp lesen Sie bitte auch im Kapitel „Tipps und Tricks“ weiter.

## **Amica Paint**

Format: Multicolor, gepackt

Lader: ja Saver: ja

Anzeige: Amica P, 160x200, Color

Dieses Format entspricht vollständig dem Koalaformat, es ist nur zusätzlich gepackt und trägt am Ende eine 256 Bytes lange Tabelle für Farbtrotationen. Außerdem hat es eine andere Startadresse. Die eckigen Klammern im Namen stellt GoDot als „ö“ und „ä“ dar.

## **BasicText**

Format: Basicprogramm aus PRINT-Anweisungen oder REM-Kommandos

Lader: ja, Saver: nein  
BasicTxt

Anzeige: Text, 320x200, Color

Der Idr.BasicTxt verwandelt a) ein lauffähiges Basicprogramm, das den Bildschirm beschriften würde (also aus lauter PRINT-Anweisungen mit darauf folgenden Textstrings besteht, z. B. PRINT „Dies ist eine Überschrift“) in Schrift, die direkt in die aktuelle, gerenderte Grafik eingebettet wird. Basicsteuerzeichen zur Positionierung berücksichtigt der Lader.

Zum anderen kann man b) ein Basicprogramm schreiben, das für GoDot kodierte Anweisungen über Farbe und Zeilenausrichtung enthält. In diesem Fall steht der auszugebende Text hinter einer

REM-Anweisung. Nach einem einleitenden Anführungszeichen folgen zunächst drei Ziffern, die sich so verstehen:

Ziffer 1: Ausrichtung      0= linksbündig (Vorgabe)  
                                  1= rechtsbündig  
                                  2= zentriert

Ziffer 2: Vordergrund      0 bis F (Vorgabe=1)

Ziffer 3: Hintergrund      0 bis F (Vorgabe=transparent)  
                                  (transparent=alles außer 0-F)

Darauf folgt ein Doppelpunkt. Er hat den Zweck, einmal gesetzte Codes nicht jedesmal wieder neu setzen zu müssen. Man kann also entweder schreiben: 1 REM "220:Dies ist eine Überschrift" (für zentriert, rot auf schwarz) oder 2 REM ":Dies ist Zeile 2" (mit gegenüber vorher unveränderten Attributen). PRINT und REM können beliebig gemischt sein. Es wird der GoDot-Zeichensatz verwendet (enthält Umlaute). Die Zeilennummern entsprechen den Bildschirmzeilen.

## **Blazing Paddles**

Format:                      Multicolor, ungepackt (40 Diskblocks)

Lader:                      ja                              Saver:                      nein

Anzeige:                    BIPad, 160x200, Color

Bilder dieses Malprogramms beginnen standardmäßig mit dem Kürzel „PI.“

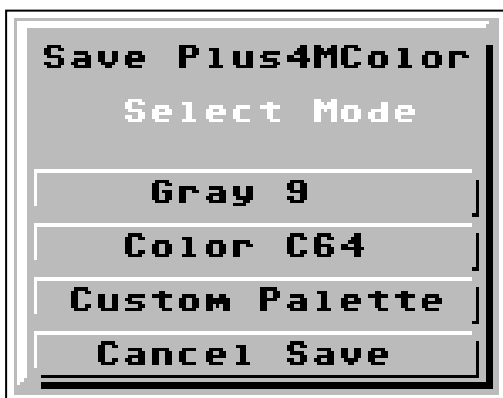
## Botticelli (Plus4)

Format: Multicolor, ungepackt (40 Diskblocks)

Lader: nein Saver: ja,  
Plus4MC16  
Plus4MC121

Anzeige: -

Der `svr.Plus4MC16` bietet mehrere Möglichkeiten, ein Bild zu optimieren. Man kann es mit den C64-Farben abspeichern (*Gadget Color C64*), aber auch mit Farben einer selbst zusammengestellten Palette. Diese Palette kann man auf dem Plus4 mit dem Utilityprogramm „P4PMaker“ bequem erstellen und auf C64-Seite mit dem Saver aufrufen, bevor man das Bild speichert (*Custom Palette*). Die dritte Option ist das Speichern in neun echten Graustufen (*Gray 9*).



Der zweite Saver kann nur sinnvoll eingesetzt werden, wenn man vorher 256-Farben-PCX-Bilder geladen und für die Speicherung als Plus4-121-Farben-Bild vorbereitet hat (*ldr.PCX-prep4Pl4*). Das resultierende Multicolor-Bild enthält bis zu 121 verschiedene Farben, die der Plus4 auch darstellen kann.



## Degas (Atari)

Format: Hires, 640x400, ungepackt (127 Diskblocks)

Lader: ja, Saver: nein  
Degas.PI3

Anzeige: Degas, 320x200/160x200, Color

PI3-Bilder sind monochrome Bilder. GoDot resamplet sie auf C64-Größe (320x200). Zwei Optionen stehen zur Auswahl (Vorrequisiter): Mit *Hires* erhält man ein hochaufgelöstes Bild in fünf Graustufen (320x200), das in der Mode-Einstellung „Multi“ hervorragend zur Wirkung kommt. Mit *Multi* berechnet GoDot ein Multicolorbild in neun Graustufen (160x200). Diese Auswahl ist besser geeignet, wenn das Bild hinterher ausgedruckt werden soll oder wenn man am Monitor den Farbreger ganz auf null dreht.

## Diashow

Format: Multicolor, Hires oder Text, gepackt

Lader: ja Saver: nein

Anzeige: Diashow, 320x200/160x200/Text, Color

Das Diashowformat gehört zu dem sehr guten Grafikgrabber-Hardwaremodul „Diashow Maker“, das einmal bei Rex-Datentechnik zu kaufen war. Das Modul hält wirklich alles fest, was es auf dem Bildschirm zu sehen gibt, auch Sprites. GoDot unterstützt alle Diashow-Bildformate bis auf die Spritedarstellung. Auch Diashowbilder mit einem Selbststarthead werden ordnungsgemäß verarbeitet.

## Doodle

Format: Hires, ungepackt (37 Diskblocks)  
Lader: ja Saver: nein  
Anzeige: Doodle, 320x200, Color

„Doodle“ ist eins der wenigen guten farbigen Hiresmalprogramme. Bilder beginnen im Namen immer mit „DD“.

## Flip

Format: FLI, gepackt/ungepackt (68 Diskblocks)  
Lader: ja, Saver: ja,  
Flip FlipRaw  
Anzeige: FLIP, 160x200, Color

„Flip“ heißt „FLI-Paint“ und ist dem Multicolor-Malprogramm „Amica Paint“ nachempfunden. Es bietet für das Malen der viel farbenprächtigeren FLI-Bilder den Komfort eines guten Standardmalprogramms.

## Funpaint II

Format: IFLI, gepackt/ungepackt (133 Diskblocks)  
Lader: ja, Saver: ja,  
IFLIprep4PCX FunPaintII  
Anzeige: IFLI, 160x200, Color

„Interlaced FLI“ bringt noch mehr Farben auf den C64-Bildschirm als FLI, allerdings zu dem Preis, dass viele der erzeugbaren Farben sehr stark flimmern, um nicht zu sagen: flackern. Eine ganze Reihe von Farben ist sich auch zu ähnlich, als dass man sie am Bildschirm auseinanderhalten könnte. GoDot beschränkt sich daher auf rund 55 deutlich unterschiedliche Farben, von denen nur wenige die Augen belasten (lässt sich mit *mod.RasterIFLI* abmildern). Soviel zum Saver.

Der Lader erkennt sowohl die Bilder des Programms „Funpaint II“ als auch die des verbesserten Nachfolgers „Gunpaint“. Funpaintbilder beginnen im Namen mit zwei Hochpfeilen.

Im 256-Farben-PCX-Format gespeichert, kann man C64-IFLI-Bilder sehr gut auf Fremdrechnern weiterverarbeiten. Will man auf dem C64 auf die besondere IFLI-Farbdarstellung verzichten, sollte man die Bilder mit den Bearbeitungsmodulen *mod.OddSwap* oder *mod.MCScreenFix* oder beiden vorbehandeln.

## GeoPaint

Format:	Hires 640x720, gepackt		
Lader:	nein	Saver:	ja
Anzeige:	-		

GoDot speichert ein 640x400 Pixel großes monochromes Bild im VLIR-Dateiformat von GEOS. Beim Saven sind die Einstellungen des Fensters „Color Controls“ aktiv. Besonders die Wahl des Ditherings hat großen Einfluss auf die Qualität des Ergebnisses.

Wenn man im Saver „Clip“ in Screen Controls einstellt, speichert er das angezeigte Bild im Format 320x200 Pixel auf einem 640x208-Pixel-Canvas.

## **GIF (MS-DOS)**

Format: Multicolor/Hires, gepackt

Lader: ja Saver: ja

Anzeige: GIF xxx, 320x200/160x200, Color

Das GIF-Format ist in Mailboxen weit verbreitet, da es Daten besonders effektiv packt. Es gibt für GoDot sowohl Lader als auch Saver. Der Lader erzeugt wahlweise Hires- oder IFLI-Bilddaten.

## **Handyscanner**

Format: 4Bit 640x512, effektiv 640x400

Lader: ja Saver: -

Anzeige: HScan, 320x200, Gray („GoDotScan“)

Dieser Lader steuert unmittelbar die Hardware des Scantronik Handyscanners an. Ein Klick auf „Load“ im Commands-Fenster startet augenblicklich den Einlesevorgang. Wenn keine Hardware angeschlossen ist, kann man mit STOP das Scannen (unter Verlust des vorherigen Bildes) abbrechen.

Der Lader erwartet Motive mit einer Fläche von real 8x5 Zentimetern im Querformat. Er liest dabei immer mit der höchsten Skalierung, die das Gerät bietet (1:1), was einer Einstellung von 300% bei der Originalsoftware entspricht. GoDotgescannte Bilder wirken am besten in Multicolor bei einer Farbtiefe von 5 mit der zugehörigen Defaultpalette (Colors: 5; Palette, Default).

Hochkant gescannte Bilder kann man mit Savern, die ein „90“ im Namen führen (Drehung um 90 Grad nach rechts) weiterverarbeiten.

## Hires Bitmap

Format: Hires, ungepackt (32/33 Diskblocks)

Lader:	ja, HiBitmap PFoxSelect	Saver:	ja, Bitmap
--------	-------------------------------	--------	---------------

Anzeige: HiBitmap, 320x200, Color

Ldr.HiBitmap hat einen Compose-Teil, mit dem mehrere Bitmaps verknüpft werden können. Er wird bedient wie der entsprechende Teil im *ldr.4BitGoDot*. Nachgeladene Bitmaps kann man auch einfärben, wenn man zuvor die gewünschte Farbe im Paletterequester anwählt.

Bilder, die mit *ldr.PFoxSelect* geladen wurden, müssen danach invertiert werden (*mod.Negative*), da dieser Lader davon ausgeht, dass die Daten im typischen Scanntronic-Look vorliegen (schwarz auf weiß).

## Hires Manager

Format: FLI Hires, gepackt/ungepackt (65 Diskblocks)

Lader:	ja, HiManRaw	Saver:	ja, HiManRaw
--------	-----------------	--------	-----------------

Anzeige: HiresM, 320x200, Color

Auch für den Hires-Mode gibt es ein FLI-Malprogramm, den „Hires Manager“. GoDot versteht nur die ungepackten Bilder. Man erkennt sie daran, dass ihr Name mit den Zeichen „(^)“ beginnt.

## IFF (Amiga)

Format: Hires, gepackt

Lader:	ja, IFF IFFdither	Saver:	ja, IFF
--------	-------------------------	--------	------------

Anzeige: IFF xxx, 320x200, Color/Gray

Bis auf die 256-Farben-AGA-Bilder und natürlich Truecolorbilder kann GoDot mit jedem Amigabild etwas anfangen. Zu große Maße (bis 1024x768) werden auf C64-Größe zurück-resizet (nicht resamp-let), wobei einige Details verloren gehen können.

Amigabilder lassen sich für höhere Detailtreue in schwarz/weiß als Graustufenbilder einlesen (Option *Gray Scale*), in Color geladene Bilder werden dagegen so gut es geht an die festen 16 Farben des C64 angepasst.

Sollten zu viele Flächen grau bleiben, kann man mit den fünf Farblevels versuchen, unbunte



Stellen beim nochmaligen Lesen doch noch farbig zu bekommen. Je höher der Level (*Lvl*), desto mehr verstärkt GoDot schwach vorhandene Tönungen. Einstellungen im *Balancing*-Requester beeinflussen ebenfalls die Farbigkeit des Ergebnisses.

## **Koala**

Format: Multicolor, ungepackt (40 Diskblocks)

Lader: ja Saver: ja

Anzeige: Koala, 160x200, Color

Das in Deutschland verbreitetste C64-Multicolorformat ist Koala. Fast jedes andere Malprogramm weiß mit diesem Format etwas anzufangen. Ein Koalaname ist immer 15 Zeichen lang, beginnt mit einem inversen Pik, auf das sich die Buchstabenfolge „PIC“ anschließt.

## **Magic Disk EMC**

Format: FLI, ungepackt (68 Diskblocks)

Lader: ja Saver: nein

Anzeige: MD-EMC, 160x200, Color

Ein weiterer Multicolor-FLI-Editor, dieser aus dem Diskmagazin Magic Disk.

## Mandelbrot Construction Set

Format: FLI, gepackt

Lader: ja, M.C.S. MCSdat      Saver: ja, MCSfli

Anzeige: MCS-EMC, 160x200, Color

Mit MCS hat man ein sehr komfortables Programm, um die berühmten Apfelmännchen-Fraktale im besonders interessanten FLI-Modus zu berechnen und anzuzeigen.

Dieses Programm ist wie GoDot in der Lage, Grafikdaten, die ein C64 nicht darstellen kann, so aufzubereiten, dass sie optimal wiedergegeben werden.

## Multicolor Bitmap

Format: Multicolor, ungepackt (32/33 Diskblocks)

Lader: ja, MCBitmap      Saver: ja, Bitmap

Anzeige: MCBitmap, 160x200, Color

Manche 64er-Slideshow arbeitet in 4 festen Farben, die nur ein einziges Mal gesetzt werden müssen. So spart man Diskettenplatz und kann schneller nachladen. Bilderfiles dieser Art enthalten keine Farben.



## OCP Art Studio (Advanced)

Format: Hires (Multi), ungepackt (36/40 Diskblocks)

Lader: ja Saver: nein

Anzeige: OCP Ast, 320x200 (160x200), Color

Bilder dieses Malprogramms haben 16 Zeichen lange Namen, die immer mit „PIC“ enden. „Advanced“ bedeutet, dass es sich um Multicolorbilder handelt (Kürzel: „MPIC“). GoDot erkennt automatisch den richtigen Grafikmodus.

## Pagesetter (Plus 4)

Format: Hires 440x720, gepackt

Lader: ja, PSetSelect Saver: ja, Pagesetter

Anzeige: PSetter, 320x200, Color

Pagesetter ist ein Plus4-Programm, das dem C64-Hardwaremodul „Pagefox“ nachempfunden ist und eine ähnliche Fülle an Funktionen bereitstellt. Zur Bedienung des Laders lesen Sie bitte beim Idr.PFoxSelect (Stichwort „Printfox/Pagefox“) weiter. Der Saver dreht das im C64-Speicher befindliche Bild um 90 Grad nach rechts, stellt es also hochkant, so dass das resultierende Bild 400x640 Pixel groß ist. Es wird auf der Pagesetterseite zentriert dargestellt.

## Paint Magic

Format: Multicolor, ungepackt (37 Diskblocks)

Lader: ja Saver: nein

Anzeige: PMagic, 160x200, Color

Das besondere an Paint Magic Bildern ist, dass sie trotz Multicolor-darstellung auf das Speichern des kompletten Farbrams verzichten. Stattdessen wird dieses behandelt wie ein einfaches Farbregister, dessen Inhalt das Farbram ersetzt.

## PCX (MS-DOS)

Format: Hires, gepackt

Lader:	ja, PCX-EGA PCX-VGA320 PCX-VGA320d PCXprep4IFLI PCXprep4PI4	Saver:	ja, PCX PCXmono PCXmono.90 PCX-VGA
--------	--	--------	--

Anzeige: PCX-EGA/VGA, 320x200/160x200, Color/Gray

Neben dem goDoteigenen 4Bit-Format haben wir das in der MS-DOS-Welt verbreitete PCX-Format am intensivsten unterstützt, da es auf PC- und Amigaseite von nahezu allen Grafikprogrammen verstanden wird. Es bietet deshalb eine Gewähr für gute Austauschbarkeit zwischen den verschiedenen Rechnern.

„Mono“ bedeutet „monochrom“ und erzeugt 640x400 (bzw. 400x640) Pixel große schwarz/weiße Bilder. Das EGA-Format

beschränkt sich auf 16 Farben in 320x200, kommt dem C64 also am nächsten.

VGA-Bilder enthalten maximal 256 Farben. Beim Laden müssen sie an die Verhältnisse des C64 angepasst werden. Der Idr.PCX-VGA320 geht da ganz kompromisslos vor. Er erwartet Bilder in der exakten Größe 320x200 oder 160x200 und passt die Farben so gut es geht an (beeinflussbar mit den Levels und Balancing). Die Preplader haben mehr Freiheiten und berechnen 53 (IFLI) bzw. 121 Farben (Plus4). Geprepte Bilder wirken am besten, wenn sie im vorgesehenen Grafikmodus angezeigt werden.

Der svr.PCX-VGA dient vornehmlich dazu, Bilder im IFLI-Format auch für PC-Nutzer ohne Farbverlust verfügbar zu machen.

## **Printfox/Pagefox**

Format: Hires 320x200/640x400/640x800, gepackt

Lader:	ja, PFoxResample PFoxSelect	Saver:	ja, PFox.GB Pagefox.90
--------	-----------------------------------	--------	------------------------------

Anzeige: PFox, 320x200, Color

Ldr.PFoxResample liest sogenannte „GB“-Bilder ein, Bilder mit 640x400 Pixeln Größe, wie sie mit jedem Scantronik-Malprogramm



hergestellt werden können. GoDot verkleinert sie dabei auf 320x200, aber auf eine Art und Weise, dass kaum Informationsverluste in Kauf genommen werden müssen (s. auch Degas).

Pagefoxbilder können ganz unter-

schiedliche Ausmaße aufweisen, bis hin zu 640x800 Punkten. Ein solches Bild wäre in seiner Gesamtheit unter GoDot nicht optimal darstellbar, daher haben wir den Ldr.PFoxSelect entworfen, der es erlaubt, mit *Set Clip* einen 320x200 Pixel großen Ausschnitt aus dem angewählten Bild zu bestimmen (40x25 Kacheln). Im Übersichtsfenster zeigt GoDot die Form des Bildes. Darübergelegt sieht man im Programm einen Rahmen aus Sprites, der den jetzigen Ausschnitt repräsentiert. Die Position seiner linken oberen Ecke wird hinter *x* und *y* angegeben. *Load Pic* startet den Ladevorgang. Sollte ein angewähltes Bild keinem Scantronikformat entsprechen, gibt GoDot eine Fehlermeldung aus.

Svr.Pagefox.90 dreht das Bild beim Speichern um 90 Grad nach rechts.

## Printtechnik Digitizer

Format:	4Bit		
Lader:	ja, PTDigiI PTDigiII	Saver:	-
Anzeige:	PTDigi, 320x200, Color („GoDotDigi“)		

Ein Digitizer verwandelt das Bild einer Videoquelle in ein Computerabbild. Unter GoDot stehen drei Treiber zur Verfügung, die jeweils unterschiedliche Generationen eines Gerätetyps unterstützen. Die zwei oben genannten wurden von der Firma Printtechnik gebaut. Sollte Ihnen nicht bekannt sein, welche Version Sie besitzen, probieren Sie beide Lader aus. Einer wird funktionieren.

Optimale Ergebnisse erhalten Sie, wenn Sie mit Schwarzweißkameras arbeiten, da der Farbträger bei Farbkameras leider zu Interferenzen führen kann.

Die Digitizerlader greifen ohne Umschweife direkt auf die angeschlossene Hardware zu. Ein Requester erscheint nicht.

## Scantronik Digitizer

Format:	4Bit		
Lader:	ja, ScDigitizer	Saver:	-
Anzeige:	ScTDigi, 320x200, Color („GoDotDigi“)		

Dies ist der dritte Digitizer. Er wurde von der Firma Scantronik angeboten. Zum Kaufpaket gehörte auch ein Set von drei Farbscheiben, mit denen es möglich war, farbig zu digitalisieren. Anders als bei der Originalsoftware, die 16 aus 64 Farben aussucht, und so farblich mehr oder weniger grob auflöst, unterstützt GoDot bis zu 4096 Farben, die auf einem Amiga oder PC auch angezeigt würden. Selbst der Idr.4BitRGB holt aus den drei Farbauszügen bessere Ergebnisse in 16 Farben heraus als die Scantronik-Software.

### **Starpainter 128**

Format:           Hires bis 640x720, ungepackt

Lader:           ja,                           Saver:           nein  
                  StarPntr.128

Anzeige:         StrPntr, 320x200, Color

Ebenso wie die Bilder von Degas und Printfox.GB werden Starpainterbilder resamplet. Auch Starpainter64-Bilder kann GoDot verarbeiten.

## ***Bearbeitungsmodule***

**D**er größte Teil der GoDotmodule dient zur Bearbeitung der 4Bit-daten im Speicher des C64. Sie können das Bild verbessern, verfremden, mehrere miteinander kombinieren, Teile herauslösen, die Größe verändern, Farben zufügen oder wegnehmen usw.

Ein in Klammern gesetzter Buchstabe in der Beschreibung sagt Ihnen, ob das Modul auf Graustufen wirkt (G) oder auf Farben (F).

Der andere Teil erleichtert die Arbeit mit GoDot und den angeschlossenen Peripheriegeräten. Diese „Systemmodule“ weisen zur Unterscheidung fast alle einen zweiten Punkt im Namen auf, so dass sie immer mit führendem Punkt angezeigt werden.

Ein weiterer Teil dient dazu, das Erarbeitete auch der Nachwelt zu überliefern, sprich, es zu Papier zu bringen. Das sind die Druckertreiber.

Alle Arten von Modulen werden ständig weiterentwickelt und vervollständigt. Wenn Sie wissen wollen, ob Sie mit aktuellen Versionen arbeiten, verfolgen Sie bitte die Bekanntmachungen in der einschlägigen Presse. Sollten Sie über Datenfernübertragungseinrichtungen (Modem) verfügen, können Sie auch direkt mit den GoDot-Autoren Kontakt aufnehmen, z. B. via BTX oder Internet.

## Systemmodule

### *.ChangeDir*

Das Modul funktioniert nur mit GoDot-Versionen ab v1.24 („Go-Dot!“ im Commandsfenster klicken). Es öffnet einen Requester ähnlich dem Filerequester, zeigt aber ausschließlich Subpartitionen der Floppy 1581 bzw. Subdirectories von CMD-Laufwerken an. Da das RAM-Device keine Unterverzeichnisse unterstützt, fehlt unter *Units* auch die RAM-Angabe.

Wenn im Übersichtsfenster die Namen von Unterverzeichnissen angeboten werden, lassen sie sich per Doppelklick oder *ChangeDir* aktivieren. Alle weiteren Zugriffe GoDots finden nun auf das ausgewählte Verzeichnis statt.

Möchte man das Verzeichnis wieder verlassen, ruft man in `mod..ChangeDir` das Gadget *Root* auf. Damit kehrt man zum Hauptverzeichnis zurück.

Floppydrives, die Unterverzeichnisse gar nicht zulassen, weist das Modul mit einer Fehlermeldung ab, sobald ein Verzeichnis aktiviert werden soll. Der Wechsel zwischen echten Partitionen auf CMD-Laufwerken sollte über das Kommando `DOS in mod..FileCopy` erfolgen, solange noch kein eigenständiges CMDTool existiert (zur Vorgehensweise s. CMD-Handbücher).

### *.ColorCount*

Zählt in einem Bild die vorkommenden Farben. Interessant für Bilder, die mit den PCXprep-Ladern in den Speicher geholt wurden, denn das Modul zeigt auch die neuen Farben an, getrennt von den



festen 64er-Farben. Bei Hiresbildern gilt nur die Angabe für Standard-C64-Farben, die anderen sind irreführend und treffen nicht zu.

## *.FileCopy*

Mit diesem Modul hat man die Freiheit, zwischen allen an GoDot angeschlossenen Laufwerken Files jeder Art und beliebiger Länge auszutauschen. Allein das Kopieren mit nur einem Laufwerk (und zwischendurch stattfindendem Diskettenwechsel) geht nicht, da GoDot die Daten blockweise liest und direkt aufs Ziellaufwerk wegschreibt.

Seine Oberfläche orientiert sich an bewährten Amigaprogrammen. Die linke Seite des Bildschirms ist dem Quelldrive vorbehalten, auf der rechten Seite erscheinen die Angaben zum Zieldrive.



Für beide Seiten gibt es ein Unitsfenster, in dem man das gewünschte Laufwerk durch Anklicken auswählt. Die Zahl wird daraufhin hell dargestellt und der Diskname, der freie Platz auf der

eingelegten Diskette sowie der Inhalt der ersten Directoryblocks werden ausgegeben. Ebenso für beide Seiten gibt es die Pfeilgadgets zum Blättern im jeweiligen Directory und die ?-Gadgets, mit denen man sich die Floppymeldung abholen kann, wenn einmal die LED blinken sollte.

Die übrigen Gadgets gelten nur für die linke Screenseite. Mit *Copy* wird ein ausgewähltes File kopiert, mit *Move* dagegen verschoben, d. h. es wird zunächst kopiert und dann auf dem Quelldrive gelöscht. Wenn bei einer Move-Aktion irgendein Fehler auftritt, unterbleibt das Löschen selbstverständlich.

Die zwei nächsten Gadgets erleichtern relativ häufig gebrauchte DOS-Kommandos. *Name* dient zum Umbenennen von Files (das Anzeigegadget ganz oben wird dann zum Eingabefenster) und *Del* dient zum Löschen. „Del“ kann innerhalb des Systems nicht rückgängig gemacht werden, also vorsichtig!

Bleibt noch *Dos*. Alle anderen Kommandos an die Floppys, die nicht von Gadgets ausgeführt werden (aber auch die), kann man per „Dos“ von Hand eintippen. Dazu dient die Statuszeile, die mit 32 Zeichen Breite lang genug sein sollte.

Mit *Quit* verlässt man das Modul.

### ***.FileType***

Auch nach Aufruf von mod..*FileType* öffnet sich der Filerequester. Diesmal jedoch nicht, um Files zu laden, sondern um durch Anklicken herauszufinden, um welche Art von File es sich beim ausgewählten eigentlich handelt.

Wenn man sich also nicht sicher ist, ob ein File überhaupt ein Bild beinhaltet, wird man mit diesem Modul sicherlich zufriedengestellt, denn es kennt weit über 40 verschiedene Fileformate.

### ***.FirstDrive***

Manche Module greifen sicherheitshalber als erstes auf das Laufwerk zu, von dem Sie gestartet sind (z. B. *mod..FileCopy*). Wenn Sie das nervt, weil Sie eigentlich von vornherein einen anderen Drive anwählen wollten, benutzen Sie *mod..FirstDrive*.

### ***.Histogram***

Dieses Modul zählt im ganzen Bild die Pixel durch und zeigt an, zu welcher Farbe sie gehören. Mit Hilfe dieser Information können Sie einige andere Module gezielter einsetzen, z. B. *mod.ApplyColor* oder *mod.DynaRange*.

### ***.HowMany***

*Mod..HowMany* zeigt an, wie viele Farben sich in jedem der tausend 8x8-Blöcke (Kacheln) des Bildes befinden. Besonders bei Bildern, die von Fremdformaten konvertiert wurden, können das unter Umständen weit mehr als die hardwareseitig maximal anzeigbaren vier Farben sein (eben bis zu 16).

Sollten in einem Bild beim Rendern („Display“ in „Screen Controls“) unschöne spitze Ecken auftauchen, dann war der GoDot-Renderalgorithmus durch zu viele gleich häufige Farben in einer Kachel überfordert. Beim Grafikmode „Hires“ und einer „Colors“-Einstellung größer als zwei trifft das beim C64 sehr schnell zu.

## .NetCopy

Paul Gardner-Stephens Programm „64Net“ erschien uns so nützlich und wichtig, dass wir mit mod..NetCopy eine erste Anbindung fertiggestellt haben. Auf diese Weise besitzt GoDot unmittelbaren Zugriff auf die Laufwerke eines PCs und Sie können bequem hin- und herkopieren.

NetCopy wurde im Original von PG-S programmiert (und liegt auch dem Programmpaket bei), hat von uns aber eine (nicht ganz systemkonforme) GoDot-Oberfläche erhalten. Bedient wird es so:



Größtes und wichtigstes Fenster ist die Fileanzeige in der Screenmitte. Wiederum in seiner Mitte ist das *Anwählgadget*. Files, die hier angezeigt werden, markiert man zum späteren Kopieren durch Klicken (das Kopieren wird später mit *Copy* ausgelöst). In der Liste bewegt man sich mit den *Pfeilgadgets*. *Home* und *End* bezieht sich auf Anfang und Ende der Liste.

Will man einfach alle Files markieren, klickt man *All* auf der linken Seite. *None* deselektiert alle Files, die Markierung verschwindet wieder. Die Zahl unter „none“ zeigt an, wie viele Files sich insgesamt in der Liste befinden.

Oben links neben der Fileliste befindet sich das *Units*-Gadget, mit dem Sie anwählen, auf welches C64-Laufwerk Sie sich beziehen wollen. Direkt darunter verändern Sie die Pfadeinstellung auf PC-Seite (*Path*). Wenn Sie hier klicken, müssen Sie unten im *Statusfenster* über die Tastatur eingeben, in welchem PC-Verzeichnis Sie suchen möchten. Schließen Sie die Eingabe bitte mit einem Schrägstrich („/“) ab.

Das Gadget *NCom* hat bisher keine Funktion, da das 64Net noch nicht in allen Punkten fehlerfrei arbeitet. Später sollen Sie hiermit die Möglichkeit erhalten, Kommandos an den PC-Server abzuschicken, um z. B. ein sogenanntes „D64“-Laufwerk als Quelle einzubinden (dabei handelt es sich um 1541-Disketten, die komplett als PC-File gespeichert wurden).

Das Gadget mit dem Commodore-Logo wechselt die Kopierrichtung, die vom Pfeil angedeutet wird. Mit *Ex* verlässt man NetCopy.

### ***.NewPointer***

Lädt einen neuen Mauszeiger, sofern sich einer auf der Diskette befindet (Namenskürzel „PTR.“)

Es handelt sich dabei um ein ganz normales Spriteimage von 63 Bytes Länge. Am Ende folgen in zwei weiteren Bytes die Informationen über den Hotspot, das ist die Stelle des Zeigers, die der Zeigefingerspitze entspricht, mit der man also deutet und auslöst (normalerweise zweimal Null).

### ***.PatternEd***

Das Rastermuster, das im Fenster „Color Controls“ unter „Dith: Pattern“ eingestellt wird, lässt sich mit diesem Modul beliebig verändern, sogar gänzlich neu definieren oder einem spezifischen Zweck unterordnen. Mod..PatternEd öffnet einen eigenen Requester, der eine ganze Serie von Gadgets enthält.

Oben im Auswahlgadget sehen Sie alle 16 Einzelmuster eines Rasters und ihre beigeordneten Farben. Klicken Sie dort, wenn Sie ein Muster bearbeiten wollen. Es erscheint daraufhin links im Editorfenster (als Vergrößerung des 8x8-Musters) und mehrfach rechts in Originalgröße, um es besser beurteilen zu können. Nach der Auswahl ist zunächst der Dot-Modus aktiv (Punkte setzen; Anzeige oben rechts neben dem Editorfenster).



Klicken Sie ins Editorfenster und es erscheinen an dieser Stelle Punkte im Muster. Klicken Sie die Stelle ein zweites Mal, verschwindet der Punkt wieder.

Um ein gänzlich neues Raster vorzubereiten, löschen Sie ein eventuell angezeigtes Muster mit *Clr* und kopieren Sie das leere Fenster auf alle anderen mit „>All“. Mit *Fill* würden Sie das Muster vollständig einfärben.

Jeweils die letzte Aktion im Editorfenster (nicht, wenn Sie bereits „Ok“ gedrückt haben!), können Sie mit Undo zurücknehmen. *Oh no!* verwirft alle Änderungen an einem in Arbeit befindlichen Raster insgesamt.

*Ok* überträgt das Muster im Editorfenster ins Raster. Es wird daraufhin oben an seiner Position angezeigt. Wollen Sie das gleiche Muster für die nächste Position wieder benutzen (und dort leicht ändern), kopieren Sie es mit *Copy*. Wenn der Copymodus aktiv ist (Anzeige), deuten Sie einfach oben im Auswahlgadget auf die neue Stelle und das Muster erscheint dort.

*Mirror* spiegelt das Muster an der senkrechten Mittelachse, Turn an der waagerechten Achse. *Rotate* dreht es um jeweils 90 Grad nach rechts.

*Scroll* dient dazu, das ganze Muster in eine bestimmte Richtung zu rollen. Sie geben sie an, indem Sie im Editorfenster auf den Rand des Musters klicken, an der Stelle, die der gewünschten Richtung entspricht. Den Scrollmodus verlassen Sie dadurch, dass Sie in die Modusanzeige klicken. Damit stellen Sie zurück auf „Dot“.

Mit *HLin* und *VLin* setzen Sie horizontale bzw. vertikale Linien im Muster. Wie bei „Scroll“ verlassen Sie auch diesen Modus, wenn Sie ins Anzeigegadget klicken.

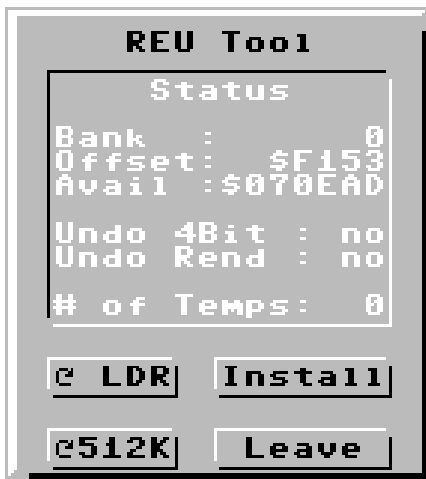
Fertige Raster sollten Sie mit *Save* abspeichern und Sie können sie dann mit *Load* jederzeit wieder in den Speicher holen. Das beim GoDotstart voreingestellte Raster erhalten Sie mit *SysPat*. Vergessen Sie nicht, eine fertige Arbeit mit *Accept* dem System zur Verfügung zu stellen, denn *Cancel* beendet mod..PatternEd, ohne die Änderungen weiterzureichen.

### ***.REUTool***

Dient hauptsächlich dazu, weitere Lader, Saver oder Module zur Laufzeit des Systems in die RAM-Disk zu installieren.

Darüber hinaus macht es einige Angaben zum Status der aktuellen RAM-Ausnutzung (Undo und Temp).

Die oben angegebenen *Bank*- und *Offset*-Adressen geben den nächsten freien Platz in der REU an. *Avail* sagt Ihnen (ohne Berücksichtigung des Tempbereiches), wieviel Platz noch verfügbar ist.



Das Gadget mit der Größenangabe (z. B. „512K“) bezieht sich auf die Gesamtgröße der REU. Sollten Sie Ihre REU mit mehr Speicher versehen haben, klicken Sie hier hinein. GoDot zeigt Ihnen Ihre Ausbaustufe an. GoDots Arbeitsbereich erstreckt sich auf höchstens ein Megabyte, auch wenn hier mehr angezeigt wird.



## *.VDCEd*

Ein Modul für C128-Freaks. Mit seiner Hilfe lassen sich eine ganze Reihe der Register des C128-Videochips VDC direkt verändern. Man kann damit seinen Monitor bis zum Rand der Leistungsfähigkeit belasten (wir hoffen, dass er nicht zerstört wird).

Chars	Dispφ	Rows	LHeight
<input type="text" value="80"/>	<input type="text" value="100"/>	<input type="text" value="25"/>	<input type="text" value="7"/>
Total	Dispφ	Total	Lace
<input type="text" value="127"/>	<input type="text" value="125"/>	<input type="text" value="38"/>	<input type="text" value="252"/>
Skip	Bitmap	Adjust	<input type="text" value="Graph"/>
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="Reset"/>
HSWidth	RAMType	USWidth	<input type="text" value="SLace"/>
<input type="text" value="9"/>	<input type="text" value="64"/>	<input type="text" value="4"/>	<input type="text" value="Leave"/>
LeftB	ChWidth	UpperB	
<input type="text" value="102"/>	<input type="text" value="8"/>	<input type="text" value="32"/>	

Interessant für alle User mit einem C128 DB wird das Gadget *SLace* sein, was so viel bedeuten soll wie „Software-Interlace-Modus“. Wenn sich ein 640x400 Pixel großes Bild im VDC befindet (*mod.4Bit2VDC2*), stellt dieses Gadget ein solches Bild 1:1 dar, also genauso wie es ausgedruckt würde.

## *.Version*

Mod..Version zeigt genauere Informationen zum gerade installierten Lader und Saver. Das wichtigste daran ist die Revisionsnummer. An ihr können Sie ablesen, ob Sie als User noch ein aktuelles Modul

benutzen. Verfolgen Sie die aktuelle Presse, um vergleichen zu können.

Als Pendant gibt es ein weiteres Versionsmodul, das als „Lader“ ausgelegt ist und entsprechende Informationen über das installierte Modul und die Grafikkenderoutinen ausgibt (ldr..Version). Wenn Sie gedruckte Informationen bevorzugen, liegt dem Sharepack 1 das Programm „pvers“ bei, das die Versionsnummern der Module einer Disk auf Papier ausgibt.

### ***MineSweeper***

Minesweeper ist ein Denkspiel, bei dem es darum geht, bestimmte Stellen des Spielfeldes nicht zu betreten (die „Minen“). Um eine Chance auf Gewinn zu haben, zeigt das Spiel an, wie viele Nachbarfelder eines bereits betretenen Feldes vermint sind. Oben links im Fenster sieht man die Anzahl der verbleibenden ungefährlichen Stellen. Ansonsten ist das Spiel selbsterklärend.

Wir haben es aus zwei Gründen programmiert. Einerseits ist ein wenig Kurzweil nie schlecht. Andererseits wollten wir zeigen, dass GoDot eigentlich eine universelle Benutzeroberfläche sein sollte, mit der man thematisch beliebige Applikationen herstellen kann, z. B. eben Spiele. Vielleicht lässt sich ja der eine oder andere Programmierer auf das Abenteuer GoDot ein?

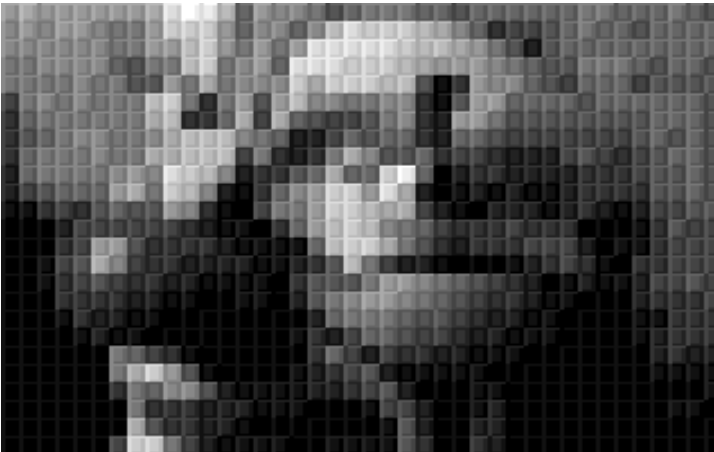
### ***READ ME!***

Wenn Sie ein READ ME- oder ein LIES MICH-File auf Ihrer Diskette finden, enthält es in Textform Informationen zum Inhalt der Diskette. So etwa beim Internet-Demo-Release von GoDot und bei den Sharepacks.

## Allgemeine Module

### *3D-MosaikMC*

(F) Jeder 8x8-Block des Bildes wird in der vorherrschenden Farbe/Graustufe gefüllt. Um den 3D-Effekt zu erreichen, werden die obere und die linke Kante aufgehellt, die rechte und die untere Kante abgedunkelt.



### *4Bit2PFox*

Rendert ein Bild direkt in das freie RAM eines Pagefoxmoduls unter Beachtung der Einstellungen in „Screen Controls“ und „Color Controls“. Nach dem Rendern kann man unmittelbar ins Pagefoxmodul überwechseln, wenn man die zugehörige Start-Software installiert hat.

Als Startsoftware taugen alle Scantronikprogramme, die ein „Fox“ im Namen führen und einen Bildschirm starten, der dem „Edison“

gleicht, also z. B. „Eddifox“, „Handyfox“ oder „Digifox“. Sie kopieren sie einfach auf die GoDotdiskette, benennen sie dort um in „gofox“ und schon geht's. Lassen Sie sich nicht von den etwas ungewöhnlichen Begleiterscheinungen beim Überwechseln irritieren. Da „gofox“ an die Stelle geladen wird, an der sich der GoDot-Zeichensatz befindet, scheint es so, als ob GoDot abstürzt. Warten Sie nur lange genug. GoDot lebt weiter!

Mit C=V kehren Sie zurück nach GoDot. Sie sollten von dieser Seite aus Ihr Pagefoxbild allerdings vorher speichern, da es dabei verloren geht.

### ***4Bit2VDC1(2)***

Rendert ein Bild unter Beachtung der Einstellungen in „Screen Controls“ und „Color Controls“ in den Videospeicher eines C128 mit 16 KByte Bildspeicher (Version mit der 1 hinten), bzw. mit 64 KByte Bildspeicher (mit der 2).

Im ersten Fall kann man das Bild auf einem angeschlossenen 80-Zeichen-Monitor als 640x200-Punkte-Bild sehen. Im anderen Fall als 640x400-Bild. Leider ist uns eine verzerrungsfreie Anzeige in 640x400 (außer bei *mod..VDCEd*) noch nicht zufriedenstellend gelungen.

### ***4BitLens***

Zeigt einen 40x24 Pixel großen Ausschnitt des 4Bitbildes in achtfacher Vergrößerung. Jedes (!) Pixel wird in seiner Originalfarbe dargestellt. Auch die, die vom Renderalgorithmus GoDots ausgelassen werden, weil etwa zu viele Farben in einem Pixelblock von 8x8 gefunden wurden.

Eignet sich gut, um Bilder, die für den IFLI-Modus aufbereitet werden sollen, vorher zu begutachten und zu beurteilen, da GoDot den IFLI-Modus nicht anzeigen kann.

Einige Tasten erleichtern das Manövrieren im 4Bitspeicher. Scrollen mit den *Cursortasten*, *Home* = linke obere Ecke; „\*“ = Bildmitte; „^“ = Anzeige des ganzen Bildes (vorher rendern); *STOP* = Abbruch des Moduls (eventuell zweimal drücken).

### ***ApplyColors***

(F) Weist die 16 Farben der Palette den Werten im 4Bitspeicher zu. Auf diese Weise kann man ungünstige Farben verschwinden lassen oder mehrere Farben auf einmal bleibend umdefinieren.

### ***ApplyDither***

(G) Reduziert das Bild im 4Bitspeicher auf die vor „Cols“ eingestellte Farbanzahl, unter Beachtung des Grafikmode und des Ditherings, und benutzt dazu die Farben der aktuellen Palette.

### ***ApplyMap(Col)***

(G/F) Überträgt die Einstellungen des Balancingrequesters bleibend auf die Bilddaten.

### ***AverageColor***

(F) Fasst zwei nebeneinanderliegende Farbwerte zu einem doppelten zusammen, dessen Wert einer C64-Farbe entspricht, die der Mischung am nächsten kommt.

## *AverageGray*

(G) Fasst zwei nebeneinanderliegende Grauwerte zu einem doppelten zusammen, dessen Wert dem arithmetischen Mittel der Einzelpunkte entspricht.

## *ClipWorks*



Wird beim Starten von GoDot standardmäßig vorinstalliert. Dient zum Manipulieren des Bildausschnitts, der in der Folge gerendert werden soll und bezieht sich auf das Gadget „Exec Area“ im Fenster „Screen Controls“.

Ein Clip ist mindestens 8x8 Pixel groß (eine Kachel), höchstens jedoch so groß wie der Bildschirm (40x25 gleich 1000 Kacheln). Er wird visuell eingestellt über das Gadget *Clip*, das den gerenderten Screen anzeigt. Dort legt man durch Klick die linke obere und rechte untere Ecke des Clipbereichs fest. Die Daten werden im Cliprequisiter daraufhin angezeigt

(hinter *Row*, *Col*, *Hor* und *Ver*, was für Zeile, Spalte, Breite und Höhe steht). Wenn man die gewünschten Daten im Kopf hat, kann man sie dort auch als Zahlen eintippen.

*Show* bringt den ausgewählten Clip noch einmal blinkend auf den Bildschirm.

*Last* rekonstruiert den zuletzt eingestellten Clip, wenn z. B. zwi-  
schendurch „Exec Area“ auf „Full“ gestellt wurde. *Full* dagegen  
macht genau das Gegenteil, ein festgelegter Clip wird endgültig ver-  
worfen.

Die Funktionen *Zoom* und *Shrink* erlauben es, einen Bildausschnitt  
auf volle Bildausdehnung zu vergrößern („Zoom“) bzw. den ganzen  
Bildschirm auf Clipgröße zu schrumpfen. Übergangswerte berechnet  
GoDot als Grauwerte. Sollten Sie lieber mit echten Farben arbeiten,  
verwenden Sie zum Strecken und Stauchen lieber die Module  
*mod.Squeeze2Clip* und *mod.StretchClip*.

*ClrClip* löscht das Innere (*Inside*) respektive das Drumherum (*Outs-  
ide*) des eingestellten Clipbereichs in einer wählbaren Farbe.

## ***DynaRange***



(G) Wenn in einer Farb-  
verteilung die dunkelsten  
oder hellsten Far-  
ben (Grauwerte) nicht  
vorkommen, wirkt das  
Bild wahrscheinlich  
recht flau. *DynaRange*  
verteilt die vorhande-  
nen Werte über den

ganzen möglichen Palettenbereich und erhöht so den Kontrast im  
resultierenden Bild.



### ***EDgray(MC)***

(G) Benutzt zur Farbreduktion und Aufrasterung ein sogenanntes Error-Distributionsverfahren (Fehlerverteilung, ED). Es erzeugt ein unregelmäßiges Muster, hat aber den Vorteil, dass es die in der Palette eingestellten Grauwerte angemessen berücksichtigt und die Dynamik des Bildes so gut als möglich zu erhalten versucht. Allerdings sollte man das System nicht überfordern und ihm zumindest Schwarz und Weiß als Bestandteil der Palette zur Verfügung stellen.

### ***Edmono***

(G) Verwendet keine Gammakorrektur, es reduziert auf Schwarz und Weiß.

Beim Ausdruck in zwei Farben machen oft die mittleren Grautöne Probleme. Oft verschwimmen sie ineinander, eine Erhöhung des Kontrastes lässt aber besonders helle und dunkle Farben verschwinden. Bei der Darstellung in Schwarz/Weiß am Bildschirm entsteht der gegenteilige Effekt, Weiß überstrahlt stark. Durch eine Gammakorrektur, die nur auf die mittleren Grauwerte wirkt, kann man per Fehlerverteilung diese Nachteile mindern.





*Edmono-0.35*

(G) Niedrige Gammakorrektur zur Druckausgabe.



### ***Edmono-1.8***

(G) Hohe Gammakorrektur zur Bildschirmausgabe.



### ***FlickerFixer***

(F) Versucht bei für das IFLI-Format ungünstigen Farbkombinationen durch Erzeugung eines Rasters Flimmern zu vermeiden.

### ***FrameClip***

Zieht einen Rahmen in einer wählbaren Farbe um den Clipbereich.

## *Gradient*

(G) Gradient erzeugt in einem Arbeitsgang ein Backdrop, d.h. ein als Hintergrund für weitere Bilder dienendes Bild. Abhängig von den Einstellungen der Fenster „Screen Controls“ und „Color Controls“ wird ein senkrechter Verlauf von der ersten bis zur letzten Farbe der Palette generiert. Im Zusammenspiel mit den anderen Operatoren und Kompositionsfunktionen der Lader kann man z.B. spektakuläre Titel erstellen.



## *MCScreenFix*

Vertauscht in jedem zweiten Byte die Nibbles der Grauwerte.

## *Mirror*

Dreht das Bild um die senkrechte (Y-)Achse.

## *Negative*

(G) Invertiert die Grauwerte aller Punkte.



## *OddSwap*

Tauscht in jeder zweiten Zeile die Nibbles der Grauwerte.

Mod.Oddswap und *mod.MCScreenFix* verbessern Bilder, die für FLI oder IFLI vorbereitet wurden, aber mit den normalen C64-Mitteln (Multicolor) dargestellt werden sollen. Außerdem entstören sie Bilder, deren Originale bereits gerastert waren.

## *Quick4(TED)*

Rendert ein Bild mit hoher Geschwindigkeit in 4 Graus. Zum Anschauen *Redisp* im Commandsfenster betätigen.

Die TED-Version ist für Bilder wichtig, die mit *ldr.PCXprep4PI4* geladen wurden, da wegen der 121 Farben nur hiermit angezeigt werden kann, wie das Bild eigentlich aussieht (Daten liegen als 7Bit-werte vor).

### ***Rendered2Raw***

(F) Wichtig für alle Saver, die ein Fremd- bzw. Nicht-Standard-C64-Format ablegen. Wenn ein Bild irgendwie behandelt wurde (Sie haben komponiert o. ä.), muss der 4Bitspeicher aktualisiert werden, denn Kompositionen finden ausschließlich in der Anzeige statt. Mod.Rendered2Raw überträgt das Bild der Anzeige in den 4Bitspeicher.

### ***ScrollR8***

Rollt das ganze Bild um 8 Pixel nach rechts und schiebt die verlorenen Bildpunkte links wieder hinein. Da bei der FLI-Darstellung die linken drei Blockzeilen nicht darstellbar sind, kann man so die relevanten 296 Pixel in die richtige Position bringen.

### ***Squeeze2Clip***

(F) Anders als die Funktion „Shrink“ in *mod.ClipWorks* verkleinert dieses Modul ein Bild und wählt für Stellen, die ihre Farbdichte dabei ändern, nicht Graustufen, sondern Farben.

### ***StretchClip***

(F) „Zoom“ in *mod.ClipWorks* vergrößert einen Clip unter Verwendung von Grauwerten. Mod.StretchClip sucht für den gleichen Zweck geeignete Farben aus.

## ***Tile***

(G) Clips können nur blockweise gesetzt werden. Eine exakte Viertelung des Bildes ist so nicht möglich. Mit diesem Modul kann man das Bild auf ein Viertel verkleinern und es dann zweimal unter- und zweimal nebeneinander darstellen. Zusätzlich kann eine Punktspiegelung an der Bildschirmmitte erzeugt werden. Besonders nach mehrfacher Anwendung ergeben sich reizvolle Bildhintergründe für Kompositionen.

## ***TileClip***

Vervielfältigt den als Clip ausgewählten Bereich und füllt damit den ganzen Bilddatenspeicher. Dabei kann man wählen, ob a) die Mitte des Clips an der Mitte des Bildschirms ausgerichtet wird oder seine rechte untere Ecke (*Snap to*), oder ob b) in X- oder Y-Richtung gespiegelt werden soll (*Mirror*). Gut für die Magic-Eye-Module von Peter Steinseifer (vor allem *mod.SIS*; hier nicht beschrieben) und für das Erzeugen von Hintergründen für Bildkompositionen.

## ***Tint***

(F) Versucht, alle Farben mit einer ausgewählten zu mischen. Auf Grund der wenigen zur Verfügung stehenden Farben des C64 ist der Erfolg leider begrenzt.

## ***UpsideDown***

Lässt das Bild um 180 Grad um die Z-Achse rotieren und stellt es so auf den Kopf.

### ***VSmooth(Col)***

(G/F) Führt dieselbe Operation wie *mod.AverageGray* senkrecht durch, aus untereinanderliegenden Punkten werden Mittelwerte berechnet.

## Konvolutionsmodule

Konvolution bedeutet „Drehung“. Das bezieht sich auf die Art, wie unser Gehirn mit den Augen die Umgebung wahrnimmt. Es erkennt zuerst grundlegende Formen (rechter Winkel, gerade Linie, Bogen) und setzt dafür fest eingestellte Detektoren ein. Je nach dem Winkel (der „Drehung“) der einen Form zu den anderen wird dann ein Gegenstand eingeordnet und als solcher registriert. Das Gehirn arbeitet dabei in höchstmöglicher Arbeitsverteilung parallel.

Eine technische Konvolution per Computer reduziert nun diesen Gesamtvorgang auf eine einzige Teilfunktion, z. B. „Erkenne eine senkrechte Linie“ oder „Erkenne Eckpunkte“. Erfunden wurden solche Untersuchungsmethoden im Zusammenhang mit der Erforschung des Weltalls und in der Medizintechnik.

GoDot bietet eine ganze Anzahl Konvolutionsmodule, mit denen Bildern völlig neue Aspekte abgewonnen werden können.





### ***Blur High***

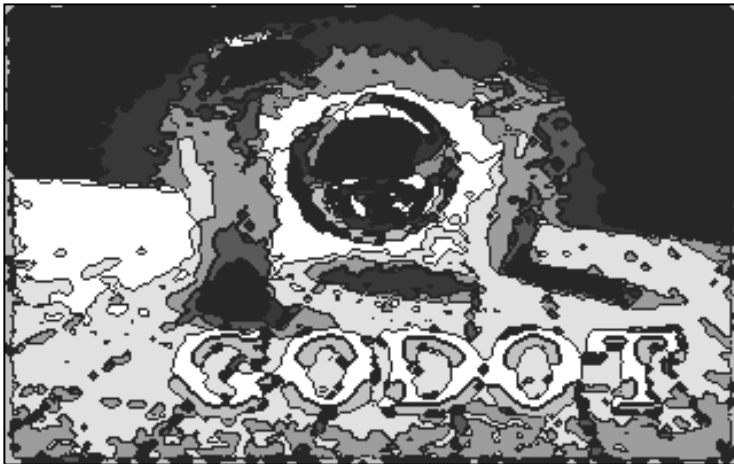
(G) Starke Unschärfe erzeugen.

### ***Blur Low***

(G) Unschärfe erzeugen.

### ***Cartoon***

(F) Kombiniert echtes Convolving mit einem Histogramm und zeichnet erkannte Flächen in der vorherrschenden Farbe, Kanten werden schwarz gezeichnet.



### ***CleanUp***

(F) Lässt ein Pixel verschwinden, wenn es keinen Nachbarn gleicher Farbe hat.

## CrossHatch

(G) Erzeugt eine leichte Struktur, die das Bild auflockert

## Convolve

(G) „Convolve“ ist das englische Wort für „Konvolution“. Dies Modul dient dazu, 3x3-Konvolutionsmatrizen zu erstellen. Die abgebildete beispielsweise durchforstet das Bild nach senkrechten Kanten und lässt nur die im Bild zurück. Wäre die mittlere Position der Matrix auch noch mit einer „1“ besetzt, würde das Originalbild wieder ins Berechnungsergebnis einfließen, und man hätte im Endeffekt eine Erhöhung der Bildschärfe. Für interessante Bildveränderungen immer gut, muss man allerdings viel mit dem Modul experimentieren, um zu wirklich einsetzbaren Werkzeugen zu kommen. Es lohnt sich!



Neue Werte tragen Sie von Hand in das Gitter ein. Sie klicken eine Zelle an, worauf dort ein Eingabecursor erscheint. Sie können Zahlen zwischen -128 und +127 eingeben.

Das Gadget *Shift* hebt alle 4Bitwerte um den angegebenen Betrag an und lässt sie dann in die Konvolution einfließen. *Threshold* richtet

eine Schwelle ein, unterhalb derer die Werte unangetastet bleiben. *Mix* legt den Grad fest, mit dem das Konvolutionsergebnis mit dem Ausgangswert verknüpft wird. „100%“ bedeutet, dass der Originalwert vollständig ersetzt wird.

Wenn man das Gadget *RotR* betätigt, rollt die Matrix im Uhrzeigersinn um den Mittelpunkt herum. Das erleichtert geringfügige Veränderungen. *Reset* setzt die Matrix auf null.

Matrizen kann man abspeichern (*Save*) und wieder laden (*Load*), wobei der Name der nachgeladenen Matrix oben angezeigt wird.

*Apply* startet den Konvolutionsalgorithmus.

### ***Convolve7Bit***

(G) Das *mod.Convolve* für Bilder, die mit dem *ldr.PCXprep4PI4* geladen wurden. Solche Bilder enthalten bis zu 121 Farben, die nur in 7 Bit kodiert werden können.

### ***DeepPress***

(G) Bei *mod.DeepPress* sehen die Bilder hinterher aus wie in Metall eingedrückt.

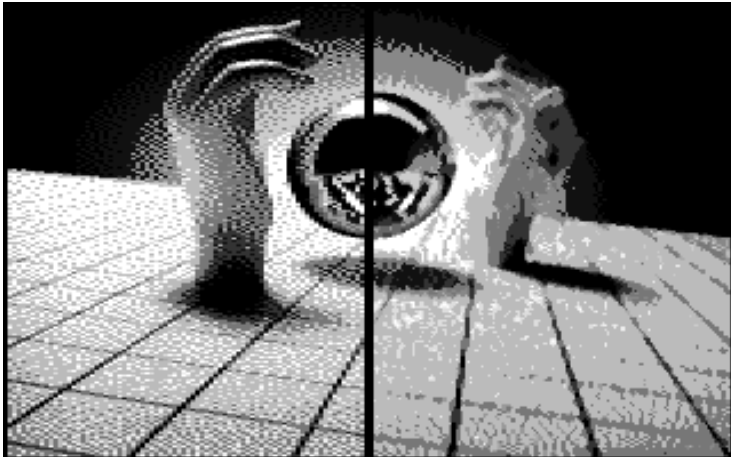


### ***Emboss***

(G) Relieffeffekt, lässt das Bild aussehen, als ob es aus Metall getrieben wäre.

### ***Gaussian***

(G) Weichzeichnen, nicht so intensiv wie bei den Blur-Modulen.



### ***LaPlacian***

(G) Eine Kantenerkennung; die Farben der Flächen verschwinden.

### ***MedianFilter***

(F) Benutzt man zum „Entrauschen“ von Bildmaterial. Die Umgebungswerte eines Punktes werden nach Häufigkeit sortiert und

genau die Farbe gewählt, die den mittleren Platz in dieser Rangliste einnimmt. Was sehr willkürlich erscheint, ist in der Praxis ein effektives Werkzeug zur Vorbereitung anderer Effekte.

### ***MedianFilterTED***

(F) Wie *mod.Convolve7Bit* ein Modul, das nur sinnvoll bei Bildern anzuwenden ist, die mit *ldr.PCXprep4PI4* geladen wurden. Hat die gleiche Aufgabe wie *mod.MedianFilter*. Anschauen mit *mod.Quick4TED*.

### ***MosaikMaxi***

(F) Ersetzt den aktuellen Bildpunkt durch die hellste Farbe der Umgebung. Durch die „grobe“ Auflösung des C64 entstehen nach mehrfacher Anwendung mosaikartige Gebilde. Recht schöner Effekt.

### ***MotionBlur(Col)***

(G/F) Verwischt Bildpunkte nach links und rechts, abhängig von der Breite des aktuellen Clips (einstellen mit *mod.ClipWorks*).

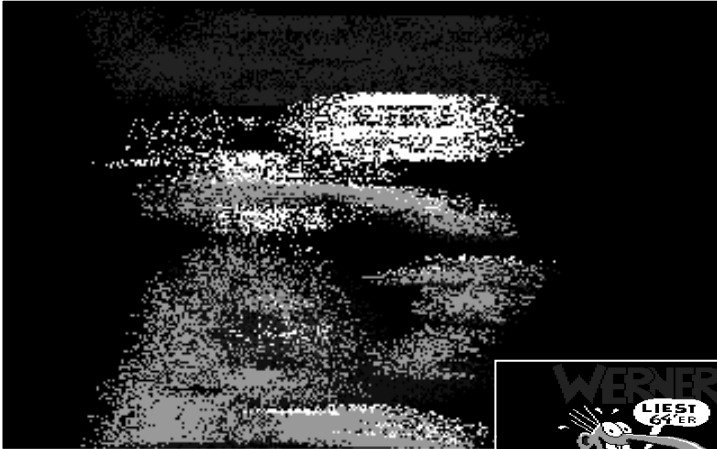
### ***OutlineColor***

(F) Wie *mod.LaPlacian*, aber die Kanten bekommen die Farbe der Fläche. Sehr detailreiche Motive bringen weniger interessante Ergebnisse.

### ***OutlineMono***

(G) Wie *mod.LaPlacian*, hier werden alle Kanten weiß. Kann zusammen mit *mod.Negative* auch zum Erstellen von Masken verwendet werden.

### ***RandMoveLeft***



(F) Verschiebt nach Zufall Bildpunkte bis zu einer definierbaren Entfernung nach links und ist bei ausreichender Wiederholung in der Lage, den gesamten Bildinhalt aus dem sichtbaren Fenster hinauszuwirbeln.

### ***RandOblique***

(G) Arbeitet in einem schräggestellten Bereich („oblique“) und erzeugt einen entsprechenden Wischeffekt.

### ***RemIsoPixels***

(F) Ein Pixel wird ersetzt, wenn seine Umgebung einfarbig ist.

### ***Sharpen1(3)***

(G) Erhöhung der Schärfe durch Kantenverstärkung.

### ***Smear***

(F) Vertauscht zufällig Bildpunkte und bringt so Unruhe in das Bild.

### ***Speckle***

(G) Eine Version von *mod.Smear* mit kräftigeren Auswirkungen.

### ***WaterColor(.O)***

(F) Erzielt einen Effekt, als ob man mit Wasserfarben malt, dadurch, dass die häufigste Farbe der Umgebung bevorzugt wird. Das "O" steht für "oblique" und arbeitet mit einer schrägen Matrix, um einen Pinselstrich zu simulieren. Vergleiche auch das Bildbeispiel zu *mod.Cartoon*.

## Druckertreiber

### ***24Pin.drv***

Geschrieben wurde dieser Treiber für einen Epson Stylus 800+ (von Markus Kanet). Er funktioniert mit allen Druckern, die sich „24-Nadel-kompatibel“ nennen.

Man kann – wie bei allen GoDot-Druckertreibern – unter drei verschiedenen Druckerinterfaces wählen: Wiesemann, Merlin C+ und Centronics. Der Anschluss des Druckers über ein Centronicskabel am Userport ist in jedem Fall die beste Wahl, da hier die Druckgeschwindigkeit am höchsten liegt.

Damit nicht jedes Mal an den Dipschaltern verstellt werden muss, lässt sich vom Treiber aus einstellen, ob ein Linefeed am Ende der Zeile gesendet werden soll oder nicht.

### ***9Pin.drv***

Der 9Pin-Treiber ist für alle Epson-kompatiblen Neunnadler geeignet. Er bietet zwei Ausdrucksqualitätsstufen: 80 dpi und 240 dpi. Die zweite Stufe ist vor allem gedacht für stark abgenutzte Farbbänder, denn die Maße des Ausdrucks sind in beiden Stufen gleich.

### ***CanonBJC.drv***

Zu den Einstellmöglichkeiten dieses Farbdruckertreibers lesen Sie bitte bei *mod.HP.drv* weiter. Anders als dort ist lediglich das Feature *Print Passes: SLOW*. Diese Option druckt ein Bild so aus, dass nach jeder Zeile 8 Sekunden Trockenpause gemacht werden. Damit werden Transparentfolien für Overheadprojektoren garantiert nicht verwischt.



Weiterhin lassen sich wegen der etwas kleineren Bildformate (der Canon druckt mit 360 dpi) mehr Bilder auf der Seite unterbringen: 21 Tiny, 8 Normal und/oder 3 Large. Ab Version 1.02 ist es mit der Shift-Lock-Taste möglich, Ausdrücke auch zu zentrieren.

## *HP.drv*

Ein Farbdruckertreiber für alle Drucker, die der PCL3-Druckersprache mächtig sind. Das sind vornehmlich alle Drucker aus der 500er-Serie von Hewlett-Packard, aber auch Laserdrucker und natürlich kompatible Fremdrunder.



Die Optionsgadgets sind ausgelegt für die Fähigkeiten des HP Deskjet 560C, aber umschaltbar auf die reinen Schwarzweißdrucker bzw. auf den Urfarbdrucker 500C (*Select-Gadget*). Wenn weitergeschaltet wird, passen sich einige der anderen Gadgets den Fähigkeiten der jeweiligen Selektion an.

Die Bilder werden bei fest eingestellten 300 dpi in drei Größen gedruckt: tiny (bis zu 18 Stück), norm (3) und Irge (2; *Format-Gadget*). Nach jedem Druckvorgang muss das Blatt neu eingelegt werden, da Tintenstrahler wie HP Deskjets meist Ganzseitendrucker sind.

Die Qualität des Ausdrucks können Sie auf drei Arten beeinflussen. Zuerst mit *Print Quality*, verstellbar auf low, high oder KPad (was so viel heißt wie „key pad“, also: gesteuert über die Druckertastatur).

Zweitens mit *Ink Usage*, der höchste Wert entspricht dem höchsten Tintenverbrauch. Die Werte mit einem „y“ vorne unterstützen die eingebaute Gammakorrektur des Druckers. Drittens können Sie mit *Print Passes* einen Punkt ein-, zwei- oder viermal drucken.

*Balanced black* ist gedacht für Colordrucker ohne Schwarz als eigene Farbe (DJ 500C). Es mildert etwas die Mischung der drei eingebauten Farben zu Schwarz, die leider normalerweise mehr wie oliv aussieht.

Das Gadget *Position* dient dazu, die bis zu 18 Druckpositionen direkt auszuwählen. Sie sollten hierfür einmal einen schwarzen Rahmen auf weißem Grund definieren (mit *mod.Clipworks*, Full, ClrClip weiß Inside, danach *mod.FrameClip* schwarz) und diesen an jeder möglichen Position (und vielleicht mehrfarbig) in allen Größen zu Papier bringen. Dann können Sie Ihre Textverarbeitung darauf abstellen und später bunte Bilder in Ihren Brief hineindrucken.

### ***StylusC.drv***

Der Treiber für den Epson Stylus Color hat sehr große Ähnlichkeit mit dem *Canon.drv*. Zusätzlich bietet er die Möglichkeit, die Bilder in 720 dpi Auflösung zu drucken. Der Treiber besitzt keine Tintensparmechanismen.

## Tutorien

### *Komponieren*

Verwendete Module:	Verwendete Datenfiles:
ldr.IFF ldr.4BitGoDot ldr.4Bit&Map ldr.4Bit&Mask svr.4BitGoDot (svr.Temp) svr.Koala (oder anderer) (mod..REUTool) mod.ApplyDither mod.ApplyColors	Clouds3.IFF SphGodAlpha.4Bit SphereBump1.4Bit Sphere.Mask IP-2.4Bit  Besitzer einer REU sollten die notwendigen Module mit <i>mod.REUTool</i> im RAM vorinstallieren.



Als erstes wollen wir den **Hintergrund** unseres Titels vorbereiten. Dazu wählen wir den *IFF*-Lader, aktivieren die Option *Gray Scale* und laden das Bild „Clouds3.IFF“.



Wer eine REU hat, sichert das Bild jetzt als „Undo 4Bit“, um sich das nochmalige Laden zu ersparen. (Im Filerequester Unit RAM anwählen und den entsprechenden Namen doppelklicken.)

Die 16 Graustufen des Bildes sollen auf eine reine Blautonpalette reduziert werden. Dazu führen wir den Image Operator *ApplyDither* aus. Im sich öffnenden Requester stellen wir die Optionen *Dith: Pattern, Mode: Multi, 6 Cols* ein und akzeptieren. Der Vorgang wird ausgeführt.

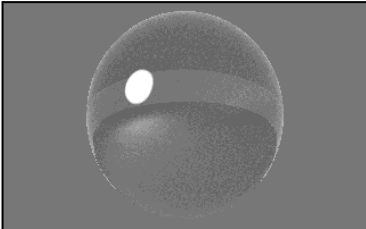
Um jetzt die richtigen Farben zu wählen, stellen wir auch im Hauptschirm unter „Screen Controls“ auf *Colors: 6* zurück. Dann wählen wir *Palette* in „Color Controls“ an, lösen *Default* aus und ersetzen in der (unten gelegenen) Palette das dunkle Grau durch Dunkelblau. Akzeptieren.

Mit dem Operator *ApplyColors* (zuerst installieren!) übertragen wir die Farben auf die Bilddaten im Speicher. Nachdem *Colors* (auf 16) und *Palette* (mit *Default*) zurückgesetzt worden sind, können wir das Zwischenergebnis betrachten: ein Wolkenhimmel in Schattierungen von Blau bis Grün.

Als nächstes müssen wir das Ursprungsbild noch einmal laden

Die REU-User laden mit dem gerade vorhandenen Lader aus dem RAM das „Undo 4Bit“.

Der **Vordergrund** wird mit dem Lader *4Bit&Map* aufgetragen. Mit dem Gadget *Modify Data* gelangen wir ins Bildbearbeitungsmenü des Laders, aus dem wir mit *AlphaCh* (Option *pos*) von der Diskette das Bild „SphGodAlpha.4Bit“ holen. Dieser Vorgang trägt ein kugelförmig verzerrtes Schriftobjekt auf den Himmel auf und dunkelt die Umgebung der Kugel ab.



Um die Kugel plastischer zu gestalten, laden wir aus dem gleichen Menü des Laders mit *Bump Map* das File „Sphere-Bump1.4Bit“. Die Kugel wird heller und erhält ein Glanzlicht.

Wieder reduzieren wir mit *mod.ApplyDither* die Farbtöne, diesmal auf rötliche 9 Farben, ansonsten mit den gleichen Einstellungen wie vorhin. Sehen wir uns das einmal an; die Farben der Kugel enthalten keine Blautöne, wir müssen nichts weiter verändern.

Nun wird's spannend: der blaue Hintergrund soll hinter die rote Kugel. Der Lader *4Bit&Mask* hilft uns dabei (installieren). Als erstes laden wir mit *Get Mask*, Option *from Disk* die Maske „Sphere.Mask“, die nichts weiter ist als eine zweifarbige Bitmap. Mit *View Mask* können wir sie sehen: weißer Kreis auf schwarzem Grund, sie maskiert die Kugel aus.

*Get 4Bit* (Option *from Disk*) lädt nun das Bild „Back.4Bit“ von der Diskette dazu, und verändert dort, wo die Maske nicht weiß ist.

Mit Option *from Temp* (bzw. *from VDC/from PFox*) macht's der REU/C128DB/Pagefox-Besitzer.

Und dann ist es soweit, wir können uns das letzte Teilergebnis ansehen. Wir verlassen den Lader (*Leave*) und rendern das Bild mit *Display* (in „Color Controls“).

Den letzten Schliff geben wir unserem Projekt, indem wir über das Bild einen vorbereiteten **Schriftzug** legen. Mit dem Lader *4BitGodot* und der Option *Foreground* im Menü *Compose* laden wir das File „IP-2.4Bit“ in den Vordergrund unserer Komposition.

Das fertige Bild kann nun zur Weiterverarbeitung im Format unseres Lieblingsmalprogrammes gesichert werden. Mit dem Koalaformat liegt man selten falsch, wer mehr sehen will, wählt FLI oder IFLI.

Wenn man ein Projekt für den Ausdruck erstellen will, sollte man beim Reduzieren nicht *Multi*, sondern *Hires* wählen, das ergibt feinere Raster. Für den Druck mit einem Schwarzweißprinter können wir sogar auf unsere oben vorgenommenen Farbreduzierungen ganz verzichten und so alle 16 Graustufen ausnutzen.

## Arbeiten mit Clips

<p>Verwendete Module:</p> <p>ldr.IFF ldr.4BitGoDot svr.4BitGoDot svr.Koala (oder anderer) mod.ApplyDither mod.Clipworks mod.FrameClip mod.Rendered2Raw (mod.Squeeze2Clip)</p>	<p>Verwendete Datenfiles:</p> <p>Leather1.IFF NageI01.IFF ColManOver.4Bit</p> <p>Besitzer einer REU sollten die notwendigen Module mit <i>mod.REUTool</i> im RAM vorinstallieren.</p>
---	---



Im zweiten Tutorium werden Sie sehen, dass die Bilddaten (*4Bit*) und die angezeigten Bilder (*Rendered*) zwei unterschiedliche Dinge sind, und diese Tatsache ausnutzen. Wieder bereiten wir zuerst den **Hintergrund** des neuen Titels vor. Dazu installieren wir den *ldr.IFF* und laden mit der Option *Gray Scale* das Bild „Leather1.IFF“.

Die 16 Graustufen des Bildes sollen auf eine Brauntonpalette reduziert werden. Dazu führen wir den Image Operator *ApplyDither* aus. Im Requester stellen wir die Optionen *Dith: Pattern, Mode: Multi, 9 Cols* ein und akzeptieren. Danach erzeugen wir das erste Zwischenergebnis mit *Display* in „Screen Controls“. Das Bild in der Anzeige ist die Grundlage für das endgültige „Kunstwerk“.

Wir sichern (mit *svr.4BitGoDot* REU-User speichern als „Undo die farbreduzierten Bilddaten als 4Bit“ in die Unit RAM. „Back1.4Bit“.

Als nächstes laden wir das Bild „Nagel01.IFF“, ebenfalls mit der Option *Gray Scale*.

Jetzt sind Fingerspitzengefühl und Vorstellungsvermögen gefragt. Unser Ziel ist es, „Nagel01“ um ca. 50% verkleinert in die linke obere Ecke zu platzieren. Mit dem Modul *ClipWorks* wählen wir über das Gadget mit der Funktion *Clip* einen entsprechend großen Zielbereich aus. Eine geeignete Clipgröße wäre etwa

Row: 3                      Col: 4                      Hor: 20                      Ver: 13

Man kann mit *Clip* die Form grob am Bildschirm aufziehen und die genauen Werte manuell (in die Zahlengadgets) eingeben. Mit *Show* überprüfen wir die Lage des Clips.



Zum Verkleinern des Bildes dient das Gadget *Shrink*. Diese Funktion arbeitet im Graustufenmodus und es dauert es eine Weile, bis das Ergebnis vorliegt. Schneller würde es im Farbenmodus gehen mit dem Modul *Squeeze2Clip*.

Nachdem wir *ClipWorks* verlassen haben, stellen wir fest, dass sich im Hauptschirm das *Areagadget* (in „Screen Controls“) von „Full“ auf „Clip“ geändert hat. Rendern mit *Display* erfolgt bei dieser Einstellung ohne Löschen des Anzeigebildschirms und nur im gewählten Bereich. Wir aktivieren also das Rendern und sehen „Nagel01“ verkleinert als **Vordergrund** vor dem unveränderten (jedoch als 4Bit nicht mehr vorhandenen) Hintergrund.

Mit dem Modul *FrameClip* verschaffen wir dem Motiv noch einen Rahmen in der Farbe Gelb, den wir mit *Display* auch sichtbar machen. Es soll aber nicht wie in den Hintergrund eingestanzt wirken, sondern über ihm „schweben“. Dazu erzeugen wir jetzt einen Schatten mit Hilfe von *ClipWorks*.

Stellen wir hierzu im Balancingrequester zunächst die Helligkeit („Brightness“) auf -4.

Zur Sicherheit speichern wir das gerenderte Bild im Koalaformat als „Back2.koala“. Oder wie gehabt im RAM, diesmal als „Undo Rendered“.

Unsere vorhin gesicherte Datei „Back1.4Bit“ (bzw. „Undo 4Bit“) holen wir zurück in den 4Bit-Speicher (laden). Mit *ClipWorks* stellen wir die Clips für den Schatten ein und akzeptieren und rendern sie jeweils sofort. Dabei muss das *Areagadget* auf „Clip“ bleiben!

Der untere Schatten bekommt folgende Einstellungen:

Ver: 2

Row: 16 Col: 6

Hor: 20

Beachten Sie die Reihenfolge: das Modul prüft auf Bereichsüberschreitung! Für die rechte Seite geben wir ein

Row: 5

Hor: 2

Col: 24

Ver: 11

Dargestelltes Bild und Bilddatenspeicher stimmen in diesem Stadium nicht überein, versuchen Sie abwechselnd *Preview* und *Redisp* (in „Commands“). Sehen Sie?

Um jetzt das dargestellte Bild zur abschließenden Behandlung in den 4Bit-Speicher zu übertragen, benötigen wir das Modul *Rendered2Raw* (installieren und ausführen).

Die freie Fläche rechts unten im Bild ist der ideale Platz für eine Beschriftung. Mit dem Lader *4BitGodot* und der Option *Compose/Foreground* laden wir das Überlagerungsfile „ColManOver.4Bit“. Unser Werk ist vollbracht.

Bevor wir das fertige Bild zum letzten Mal rendern, stellen wir im Balancingrequester die Helligkeit zurück (*Reset*) sowie das Areagadget auf „Full“.

Speichern Sie in einem beliebigen Format.

## Zum Schluss

### *Tipps und Tricks*

**Flip:** Eine Drehung des Bildes um die horizontale (X-)Achse erzielt man, wenn man die beiden Module *mod.UpsideDown* und *mod.Mirror* hintereinander ausführt.

**Masken für *ldr.4Bit&Mask* einfach erstellen:** Bilder, die deutlich voneinander abgesetzte Hauptmotive aufweisen (etwa das Beispielbild zu *mod.RandMoveLeft*) können leicht in Masken gewandelt werden. *Mod..PatternEd* aufrufen, *Fill*, *>All*, *Invert*, und die Farben, die nicht ausmaskiert werden sollen, mit *Copy* anklicken. Dann *Accept*. Im Hauptschirm Mode: Hires, Colors: 2, Dith: Pattern einstellen und mit *Display* rendern (wird blau/schwarz). Die Maske ist fertig.

**Svr.Temp bei VDC oder Pagefox:** Da für beide Ramerweiterungen kein RAM-Handler existiert (für REUs gibt's das dev.REU), kann man das RAM nicht über die Unit RAM im Filerequester erreichen. Man gibt stattdessen beim aktuellen Laufwerk den Auslösernamen „temp00“ ein und schon läuft's. Wenn VDC und REU beide vorhanden sind, kann man sogar zwei Temps anlegen!

**REU neu konfigurieren:** Manchmal möchte man den Inhalt seiner REU komplett neu aufbauen, beim Neustart ignoriert GoDot aber alle RAM-Anweisungen in der INI-Datei. Was tun? Die schnellste und radikalste Lösung: innerhalb GoDots den Filerequester aufrufen, RAM anklicken und den Resetknopf drücken. (STOP/Restore ginge auch.) Beim nächsten Booten werden alle Module gemäß der aufgerufenen INI-Datei neu installiert.

Erklärung: Das Kommando REUCLEAR=0 in der Standard-god.ini verhindert das Löschen des REU-Directories beim Neustart. Das Flag dafür sitzt *innerhalb* der REU. Bei aktiviertem REU-Zugriff tauscht GoDot RAM und Laderbereich, so dass das Flag beim nächsten Booten scheinbar nicht mehr da ist. GoDot startet „normal“. Mit REUCLEAR=1 am Anfang einer INI müsste trotzdem zweimal gebootet werden, damit die Änderung in Kraft tritt.

**Beim Speichern als Printfox-640x400-Bild kommt eine unansehnliche Ansammlung von Klötzchen heraus:** Alle Saver, die ein 640x400-Monochrombild erzeugen (*svr.PFoxGB*, *svr.PCXmono*, *svr.GeoPaint*, *svr.Pagesetter*, *svr.Pagefox.90*, *svr.PCXmono.90*, *mod.4Bit2VDC*, *mod.4Bit2PFox*, *mod.9Pin.drv*, *mod.24Pin.drv*), arbeiten mit den Einstellungen des „Color Controls“-Fensters zusammen! Vor allem die Dither-Einstellung wirkt hier wahre Wunder! Wer einen C128DB mit einem zweiten Bildschirm hat, kann sich mit *mod.4Bit2VDC2* und *mod.VDCEd* (Gadget *Slace*) ein Preview mit echten 640x400 Pixeln anzeigen lassen! (Flimmert zwar stark, aber es ist *alles* drauf.)

**Files lassen sich nicht laden:** Sollte ein File sich partout weigern, geladen zu werden, obwohl es doch offensichtlich vorhanden ist, gibt es dafür zwei Gründe:

a) Der Filename endet mit einem oder mehreren Leerzeichen. Abhilfe: File anklicken, der Name erscheint im INPUT-Fenster. Dort klicken, von Hand einen Stern hinter das letzte Zeichen setzen.

b) Das File enthält Standard-ASCII-Zeichen (z.B. bei GEOS-Namen). Die unter GoDot groß erscheinenden Zeichen sind GEOS-Kleinbuchstaben, sie sind im C64-Zeichensatz zwar vorhanden und darstellbar, aber nicht von der Tastatur aus zu erzeugen. Abhilfe: Alle Großbuchstaben durch Fragezeichen ersetzen. Im Vorfeld: In GEOS nur Großbuchstaben verwenden, die sind problemlos.

c) Das File taucht im Directory nicht auf, obwohl Sie es gerade abgespeichert haben. Grund: Sie haben an vierter Stelle im Namen einen Punkt gesetzt. Abhilfe: *mod..FileCopy* laden und die Datei umbenennen.

**FileCopy a)** Files löschen: Einzelfiles klickt man an und löscht sie mit *Del*. Mehrere Files gleichzeitig werden mit der *Dos*-Funktion per Hand gelöscht. *Dos*, *s:a\** <Return> löscht alle Files, die mit „a“ anfangen. (Das funktioniert auch im Filerequester: ins INPUT-Gadget klicken, *a\** <Return>, *Delete* klicken. Achtung! \* allein löscht alle Files mit dem angezeigten Kürzel, z. B. alle Lader!

b) Files kopieren: Wenn der Name bekannt ist (aber weit hinten im Directory liegt), irgendein File anklicken, der Name erscheint oben in der Anzeige. Dort klicken, mit *C=x* den Namen löschen und stattdessen den gewünschten Namen eintragen. <Return>, dann *Copy* anklicken.

c) Files kopieren, deren Namen auf dem Ziellaufwerk bereits existieren: Vor *Copy* die <Shift Lock>-Taste drücken.

**PCX-Bilder:** Wer Cliparts (also Schwarzweißbilder) vom PC importieren möchte, sollte darauf achten, dass sie dort nicht als 256-Farben-Bilder gespeichert sind, weil der dafür zuständige *ldr.PCX-VGA* Bilder erwartet, die die Maße 320x200 oder 160x200 einhalten. Dem *ldr.PCX-EGA* (für max. 16 Farben) sind Höhe und Breite egal. Wenn das Bild dann statt weiß nur dunkelblau anzeigt, setzt man im Paletterequester die Farbe Blau auf Weiß und ruft *mod.ApplyColors* auf.

**Das Saven klappt nicht:** Wenn GoDot beim Aufruf der *Save*-Funktion im Filerequester kommentarlos abbricht, ohne mit dem Speichern überhaupt anzufangen, rufen Sie *mod..FileCopy* auf, setzen als Zieldrive den Quelldrive, klicken irgend ein File an und wählen

*Copy*. Da Copy auf einem Laufwerk nicht geht, bricht GoDot mit „ER-ROR“ ab. Wenn Sie jetzt FileCopy verlassen, sollte Save funktionieren. Erklärung: Das Anlesen des Files bewegt den Schreiblesekopf des Laufwerks vom Directorytrack weg, danach klappt die Erkennung der Diskette wieder.

**Formatieren mit Dos in FileCopy geht nicht:** Frische oder fremdformatierte Disketten bringen ebenfalls die Diskerkennung durcheinander, wenn Sie darauf zugreifen. Legen Sie eine andere, formatierte Diskette ein, lassen Sie sich das Directory anzeigen, und tauschen Sie dann gegen die neue aus. Jetzt funktioniert *Dos*.

**Umlaute über die Tastatur eingeben:** Die Umlaute sind schreibmaschinenähnlich angeordnet, so dass man nicht lange überlegen muss. Drücken Sie für

ä:      Shift ;	ö:      Shift :	ü:      @	ß:      £
Ä:      C=a	Ö:      C=o	Ü:      Shift @	

## Die INI-Dateien

Noch eine Besonderheit, die GoDot auszeichnet und die Sie bisher nicht kennengelernt haben: Sie können GoDot bereits beim Starten mitteilen, welche Lader, Saver und Module Sie sofort zur Verfügung haben wollen. REU-Besitzer können schon an dieser Stelle ihre RAM-Erweiterung mit den später für die Arbeit erforderlichen Modulen füllen und sparen dadurch erheblich Zeit.

Sie schreiben dazu mit dem Basiceditor des C64 (als ob Sie ein Basic-Programm schreiben wollten) eine Datei, in der Sie GoDot Ihre Wünsche mitteilen und speichern sie wie ein Basicprogramm ab. Der Dateiname muss nur als Kennung mit der Buchstabenfolge „INI.“ beginnen. Innerhalb der Datei fängt jede Zeile mit einem Anführungszeichen an. Es gibt folgende Kommandos:

Befehl	Parameter	Bereich	Vorgabe	Zweck
COLORS	Zahl	2 bis 16	2	Anzahl der Renderfarben setzen
DEFCOLORS	Anzahl, Zahl, Zahl, ...	2 bis 16, 0 bis 15	16, 0, 6 ... (GoDot- palette)	Vorgabepaletten definieren
TEXTCOLORS	Zahl, Zahl, Zahl, Zahl, Zahl	0 bis 15	0, 14, 12, 15, 6	Bildschirmfarben für GoDot setzen
MOUSECOLORS	Zahl, Zahl	0 bis 15	1, 0	Mauszeigerfarben
MOUSEXY	Zahl, Zahl	0 bis 511, 0 bis 255	80, 80	Mausposition beim Start

CLIP	Zahl, Zahl, Zahl, Zahl	0 bis 24, 0 bis 39, 1 bis 40, 1 bis 25	0, 0, 40, 25	Clipgröße ein- stellen (Vor- gabe: FULL)
MODE	Zahl	0 oder grö- ßer 0	0	Grafikmode: 0 = Hires, 1 = Multi
TICKS	Zahl	0 bis 255	13	50tel Sekun- den für Dop- pelklick
LOADER	ldr.xxx	bis 16 Zei- chen	ldr.4Bit- GoDot	Lader instal- lieren
SAVER	svr.xxx	bis 16 Zei- chen	svr.4Bit- GoDot	Saver instal- lieren
MODULE	mod.xxx	bis 16 Zei- chen	mod.Clip Works	Modul instal- lieren
RAM	xxx.xxx	bis 16 Zei- chen	-	Ldr/svr/mod im RAM vor- installieren
PATTERN	pat.xxx	bis 16 Zei- chen	pat.Sys- tem	Raster instal- lieren
POINTER	ptr.xxx	bis 16 Zei- chen	ptr.Stan- dard	Mauszeiger installieren
LOADFROM	Zahl	8 bis 11	Bootlauf- werk	zuerst laden von ...
SAVETO	Zahl	8 bis 11	Bootlauf- werk	zuerst spei- chern auf ...
VDC	Zahl	0 oder grö- ßer 0	1	Nutzung des VDC: 0 = Module 1 = Grafik
PFOX	Zahl	0 oder grö- ßer 0	1	Nutzung des Pagefox:



				0 = Module 1 = Grafik
REUCLEAR	Zahl	0 oder größer 0	1	Löschen der REU beim Start: 0 = nein 1 = ja

Eine einfache INI-Datei sähe z. B. so aus (speichern Sie sie als „ini.ifli“ auf Disk):

```
100  "loadfrom 9
110  "saveto 10
120  "loader ldr.PCXprep4IFLI
130  "saver svr.PCX-VGA
140  "module mod.RasterIFLI
```

Beim Aufrufen des Startprogramms hängen Sie einfach den Namen der INI-Datei an die Eingabezeile an:

```
LOAD „godot“,8 : „ifli“
RUN
(bzw. Shift RUN/STOP hinter „ifli“)
```

oder

```
LOAD „godot“,8
RUN: „ifli“
```

Soll das ganze ohne weitere Tipperei sowieso immer geladen werden, ändern Sie einfach die vorgegebene GoDot-Startup-Sequenz namens „god.ini“. Sie wird genauso erstellt wie andere INI-Dateien.



Arndt Dettke, Wolfgang Kling

## Bilder der Zusatzdiskette

Damit Sie besser einschätzen können, welche Bilder und Bildbearbeitungsfiles Sie in den Tutorien eigentlich benutzen, finden Sie hier einen Überblick. Auch die anderen Bilder der Zusatzdisk sind gelistet, sofern nicht im Buch schon gezeigt.



Bild 1: Clouds3.IFF

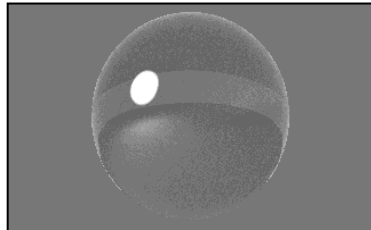


Bild 2: SphereBump1.4Bit



Bild 3: SphGodAlpha.4Bit

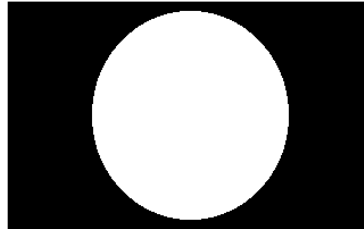


Bild 4: Sphere.mask



Bild 5: IP-2.4BIT

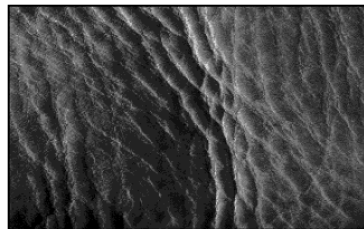


Bild 6: Leather1.IFF



Bild 7: NAGEL01.IFF



Bild 8: ColManOver.4BIT



Bild 9: castle.pic



Bild 10: bass.pic

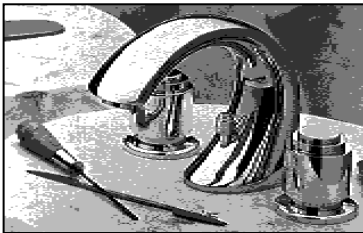


Bild 11: Chrome



Bild 12: Champagne.pcx

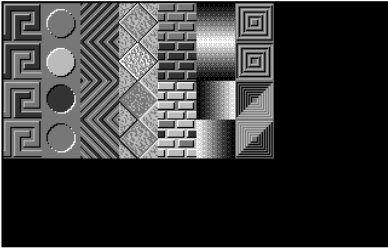


Bild 13: clippats.4bit



Bild 14: front pic pic



Bild 15: GoDotBump.4bit



Bild 16: lucky luke



Bild 17: MARLENE.gb



Bild 18: sunset.pic