

CRACK ART

THE MANUAL

CRACK ART Manual 1.36 TT+

6th Revised Edition

November 30, 1992

Copyright Detlef Röttger and Jan Borchers, 1992

Manual Design & Layout by Atarian Computing, 2020

Translation copyright Miami Valley Atari Computer Enthusiasts, 1993. Permission to copy and distribute is given as long as this notice remains with the document. Commercial sale of this translation is forbidden.

Miami Valley Atari Computer Enthusiasts
P.O. Box 24221
Dayton, Ohio, USA 45424

\$3.00 For Disk Library Listing

Translator's Note: This is a first draft and has not been edited; it may contain errors. Neither I nor the Miami Valley Atari Computer Enthusiasts will assume responsibility for any loss of income, hardware, or software which you may think resulted from the use of Crack Art. The risk is yours. The authors likewise assume no responsibility for any damage to hardware, software, or income you think may have been caused by use of the program. They have had no reports at all in four years, and they (and we) do not expect you to be the first.

Please note that most of the comments inside [square brackets] are my comments, and were not in the original docs.

Author's Note: We like to thank the Miami Valley Computer Enthusiasts and especially the translator, who was too modest to state his name here himself. Thanks for the great work and the comments, that helped us to improve the understanding of some explanations. I edited this file twice and think you will not find any further information in the German version of this manual. We hope, that this manual will help all further users, to find their way through the jungle of our weird ideas and the whole bunch of functions and possibilities of Crack Art.

Detlef and Jan

Atarian Computing's Note: Crack Art was my go-to drawing program when growing up in the 90's. A program as complex as this, deserves a detailed manual to compliment it. I have taken some artistic and linguistic liberties, changing the order of some parts, fixing spelling and some grammar mistakes. I have tried to take care to preserve the translation and the original as it was though. I also admit that this is far from perfect and I have to force myself to release this now as pursuing perfection will be to no one's benefit.

SHAREWARE ANNOUNCEMENT AND DISTRIBUTION LIMITATIONS

CRACK ART is, after long delay, now shareware and may not be sold commercially. The program may be further copied only as a complete unit (all files EXCLUDING the registration file!!). Uploading to bulletin boards is also permitted. Distribution by mail-order PD businesses is permitted as long as no or nearly no profit is made on the program or the disks. Should you receive CRACK ART via mail order, please let us know. If you use regularly or for more than 4 weeks, you are expected to send us 30 DM or equivalent (No play money, please).

The shareware fee has changed to 30 DM (or \$20 USA, æ15 UK, \$30 Canada, \$30 Australia, 100 ¥¥ France). If you send money in other currencies than DM (Deutsche Mark), we cannot accept cheques, because to cash a foreign cheque would cost more than the half of our fee. So, cash is welcome! Eurocheques in DM work fine, too. (There was a small article in the ST-Format magazine, that said, to pay the fee via an International Money Order. We never said that! The charge for this form of cheque is as big or even bigger than for any other cheque in a foreign currency. So, please do not waste money (and time, as experience showed!) with IMOs.

The pack and depack routines are PD and may be used in your own programs and further distributed as long as you put a note in the program or its documentation to the effect that the graphics were produced with CRACK ART. Also, the use of the algorithms in commercial programs is permitted as long as you include such a note and mention the copyright holders of the pack and depack routines.

One registration is good for all versions of CRACK ART. Should you as a registered user download a new version, you will have the current full version. You need only copy your personal registration file in the work index of the program.

Please register and thanks a lot to all who already did it,

Jan and Detlef

CONTENTS

Shareware Announcement and Distribution Limitations	3
Contents	4
Foreword to CRACK ART	8
File Listing	8
Updates:	9
Version 1.36 TT+	9
Version 1.35 TT+	9
Version 1.34 TT+	10
Version 1.33 TT+	10
Version 1.32 TT+	10
Version 1.31 TT+	10
Version 1.30 TT+	10
Version 1.25 TT+	10
Version 1.20 TT+	10
Version 1.12 TT+	10
Version 1.11 TT+	11
Version 1.10 TT+	11
Compatibility & Requirements	11
PR - Public Relations	11
Extract from ST FORMAT, May 1992.....	11
Extract from ST FORMAT, July 1992, Page 31.....	11
Extract from ST-Magazine, October 1992, pages 120-121.	12
Extract from ST-Computer, October 1992, page 150.....	12
Suggestions	12
Crack Art Statistics	12
Blowing our own horn.....	13
Jan 'JayBee' Borchers.....	13
Detlef "Roy" Röttger.....	14
Crack Art Overview	14
On the Construction of the Menu	15
General Overview of Alert & Dialog Boxes.	16
Alert Boxes.....	16
Dialog Boxes	16
File Select Box.....	16
SHAREWARE	16
SETS.....	16
ASM	17
Blocks.....	18
1. Cutting out a rectangular block.....	18
2. Cutting out a round block	19
3. Cutting a block with a lasso	19
4. Cutting out a block via a fill function	19
What do the various block modes have to say?	19
XRAY.....	20
OBJECT.....	20
BLOCKED.....	20
More about block modes:.....	20
Bordering the block with the screen edges.....	20
The BRUSH Functions	20
DRAW	21

POINT	21
LINE.....	21
K-LINE.....	21
FRAME.....	22
CIRCLE.....	22
ERASER	22
AIRBRUSH.....	22
RAYS	23
POLYGON.....	23
BOX.....	23
DISC	24
TEXT.....	24
Save Parameters:	25
Load E24 Font:	25
OUTLINE	25
FILL.....	26
SMEAR.....	26
MIRROR	26
SPLINE.....	27
The Working Screens	27
Functions with the Working Screens.....	28
Screen Erase:	28
Copying, Swapping, Flooding:.....	28
Add to Animation List:	28
SHOW	28
The Color Menu	29
Options for manipulating the color palette	30
Predefined Color Palettes:	30
Copying a Color to Another Register	30
Swapping Two Color Registers	30
Producing a Color Sequence	30
Determining the Color of a Point on the Screen	30
Direct Color Selection	30
Special Functions:	30
Sprite Animation.....	31
ADD - Introducing sprites into the animation sequence	32
CPY - Copy the current sprite into the one-sprite buffer	32
DEL - Erase the current sprite	32
Playing the sprite sequence	32
Block Operations.....	33
X-RAY.....	33
OBJECT	33
BLOCKED	33
FREE MOVE.....	33
EDGE STOP.....	33
SIZE	33
ROTATE.....	34
H-SKEW	34
V-SKEW	34
H-PERSP.....	35
V-PERSP	35
CURL.....	35
BULGE	35
TUBE.....	36
FIT-IN	36
DITHER.....	37
CUT	38
Other block functions in the working screen	38
BACKSPACE-key.....	38
RETURN-key	38

Cursor-keys.....	38
Clr/Home-key	38
The Magnifying Glass or Zoom	38
How Can I Call Up the Magnifying Glass?.....	39
On the Construction of the Magnifying Glass	39
On the Functions in the Magnifying Glass	40
Disk Operations	40
General.....	40
The Layout of the File Selector Box	41
Changing the file format: The PIC-TYPE menu	41
Picture Formats.....	41
EXT Origin Resolution	42
Compression	42
LOAD - (F1).....	42
Special Handling of Other Picture Formats and/or Resolutions	42
SPECTRUM 512	43
IFF Format.....	43
SAVE - (F2)	44
Conversions.....	44
SHOW - (F3).....	44
DISK MENU - (F4)	45
CRACK FILE - (F5)	46
LOAD - (RETURN Key).....	46
SAVE.....	46
VIEW.....	46
EXIT - (ESC Key)	46
SEARCH (SPACE BAR).....	47
WIDTH.....	47
OFFSET.....	47
CRACK DISK - (F6)	47
EFFECTS (FX)	47
Effects Functions	48
ANTI 4.....	48
ANTI 8.....	48
ROUND.....	48
REL M	49
REL C	49
MINI	49
RES 2	49
RES 4	50
PIXEL	50
OR.....	50
MASK.....	50
SEPARATE.....	50
OUT 4	50
OUT 8	51
EDGE.....	51
MED-.....	51
MED	51
MED+.....	51
ERO.....	51
CONTR	52
DILAT.....	52
UNZAG	52
H-SHAT	52
V-SHAT	52
USER.....	52
FADE.....	53
SWEEP.....	53
LIGHT.....	53
DARK.....	53

GREY.....	53
RED.....	53
GREEN.....	54
BLUE.....	54
CYAN.....	54
MAGENTA.....	54
YELLOW.....	54
MID.....	54
HI.....	54
TILE.....	55
H-FREE.....	55
H-SIN.....	55
H-ZIG.....	55
V-FREE.....	55
V-SIN.....	55
V-ZAG.....	56
SCREENS.....	56
SPRITES.....	56
RELAX.....	57
ASTROID.....	57
Technical Data:.....	57
Control:.....	57
Remarks:.....	58
Gameplay:.....	58
BRAIN (Master Mind).....	58
Technical Data:.....	58
Control:.....	58
Remarks:.....	58
Gameplay:.....	58
T-TRIS.....	58
Technical Data:.....	58
Control:.....	58
Remarks:.....	59
Gameplay:.....	59
PACNAM.....	59
Technical Data:.....	59
Control:.....	59
Remarks:.....	59
Gameplay:.....	59
MINE.....	59
Technical Data:.....	59
Control:.....	59
Remarks:.....	60
Gameplay:.....	60
PRINTER.....	60
SCAN.....	60
ACC.....	60
Keyboard Mapping.....	62
Key Assignments in the Menu.....	62
Keyboard Assignments in the Work Screen.....	62
Drawing Functions:.....	63
HAVE YOU EVER SEEN?.....	63
THE ULTIMATE END - YOU DID IT!.....	64

FOREWORD TO CRACK ART

In view of the worldwide distribution and the resistance to shareware fees, we have found it necessary to distribute the unregistered version of CRACK ART in a partially disabled form. Sorry, but the most of you have brought it on yourselves. The limitations are:

- two work screens (of nine possible)
- no storage (0 of 24 formats)
- after ten minutes work, the program takes a one-minute coffee break.
- periodically as you work, a message reminds you to register (we do not annoy you if you are registered).

When you register, or if you have questions of the authors, please mention the version number. This is found in the startup screen, by clicking on the SHAREWARE button, or by clicking on the Copyright in the main menu.

Registrations, programs using Crack Art, mail, phone calls, faxes, donations :-), and whatever should be sent to the following addresses: (Note: The program still contains the old postal codes.)

The German system of postcodes changed after July 1st, 1993 to five digits. Our addresses are now:

Detlef Roettger
Vienenburger Str. 27
38690 Vienenburg
Germany

Jan Borchers
Vitorwall 10
38640 Goslar
Germany

Tel/Fax: (+49) 05321/43998

Please use those addresses for quicker responses. The older ones still work but take probably one day longer. If you own a fax-machine, please let us know your fax number, because you can get your registration-code via fax a lot quicker than by post.

Telephone or fax (from the U.S.): 01149-05321-43998
(Is that correct? or should it be 01149-5321-43998? Detlef.)

File Listing

The following files comprise the complete, current version of Crack Art. Indexes are marked by <directory>.

<CRACKART>	
CRACKART.PRG	Main program
CRACKART.CAT	Title pictures
<COMPRESS>	
CA_PACK .GFA	Program example and
CA_PACK .INL	documentation
CA_PACK .LST	
CA_PACK .S	
CA_UNPAC.GFA	
CA_UNPAC.INL	
CA_UNPAC.LST	

CA_UNPAC.S	
<DEMOPICS>	
*.GIF	Demo pictures in GIF-format
<FONTS>	
*.CA1	Font pictures
*.DEF	and their definition files
<GAME>	
GAMERUN .PRG	Ther games...
ASTROID .GAM	Runtime module for games
BRAIN .GAM	
MINE .GAM	
PACNAM .GAM	
T_TRIS .GAM	
<MANUAL>	
CA136GER.LZH	German manual for version 1.36
CA136ENG.LZH	English manual
LHARC.TTP	
<PRINTER>	
<HP>	
<DJ500>	
<DJ500C>	
<NEC>	
<CP6>	
<P6>	
<STAR>	
<NL10>	
<LC100>	
<SCRAPACC>	
ACC_SCRP.ACC	Program example for the SCRAP
ACC_SCRP.GFA	Interface
ACC_SCRP.LST	

Unfortunately, some PD mail order firms have distributed only the rudimentary pieces of the program. Should you find yourself in possession of only a partial program, send the shareware fee to the authors and they will send you the complete, current version as quickly as possible. After all, they are the source.

Updates:

VERSION 1.36 TT+

- File extensions *.* and ??? remain active in LOAD (Hi, Arne!)
- Debugging

VERSION 1.35 TT+

- FILL function with dithering at arbitrary angles
- Online block size on the TT
- Minesweeper added to games

VERSION 1.34 TT+

- Horizontal centering of the block with CONTROL+CLR/Home
- Double click in the menu simulatable with ALT+left mouse button

VERSION 1.33 TT+

- Storage of color palettes as assembler source code
- Debugging

VERSION 1.32 TT+

- Storage of sprite sequences as assembler source code
- Debugging

VERSION 1.31 TT+

- Execute other programs (*.PRG, *.TOS, etc)
- Interface to other programs
- Extended effects menu
- Screen switching in color menu
- Debugging

VERSION 1.30 TT+

- Sprite animation
- Altered boot virus test
- Flexible scanner interface
- Text sizes up to 96*96 pixels
- Extended text parameters
- Debugging

VERSION 1.25 TT+

- Variable driver buffer sizes
- New title pictures

VERSION 1.20 TT+

- New effects functions with user defined filter
- Extended disk menu (added folders)
- General beautification (plastic surgery)
- Support for system text parameters 'bold' and 'light'
- Extended transfer structure for subroutines
- New game, 'Pacnam'
- Title pictures

VERSION 1.12 TT+

- New picture format, GIF (as yet only 320*200, 4 planes)
- Scanner preparation
- Automatic format recognition during picture loading

- New printer drivers

VERSION 1.11 TT+

- Scrap buffer saved and restored after program end
- Debugging

VERSION 1.10 TT+

- Menu surgery
- New game 'T-TRIS'
- Support for four screens of monochrome pictures
- Debugging

Compatibility & Requirements

Crack Art should run on all ST, STE, and TT models with at least one megabyte of RAM, as long as the memory is not too loaded with accessories and resident programs. The Falcon still rejects some routines. (That was on a demonstration model at a show. We have no idea whether it would run better now. Can someone test it for us? Known Error: ROTATE will result in 11 bombs.)

NVDI (2.0 only!?) produces problems in the zoom function which can lead to a crash. Because we do not have a copy of the program, we can only guess at the cause of the problem and hope someday to cure it. But: no guarantees! Therefore, NVDI out, CRACK ART in!

PR - Public Relations

EXTRACT FROM ST FORMAT, MAY 1992

Crack Art Version 1.00

"Public Sector - ART AND MUSIC"

ST FORMAT GOLD AWARD for Crack Art

"There are plenty of art programs knocking about in the Public Domain, but most are - er, not exactly stunning. Until recently, one of the best was the venerable Neochrome from Atari. Crack Art is the latest in a long line and knocks the competition in a cocked hat... The zoom is adjustable and fast. In fact, it's all fast - even the most complex stuff... If you use your ST to throw pixels around, then do yourself a favour and get this amazing program. Crack Art gets its name from its incredible ability to crack into programs and nick their graphics - but that's just the beginning. Underneath the smooth and silky exterior lies a very smooth program - essential for all ST pixel manipulators. The range of special effects is, to say the least, impressive - anti-aliasing, diffusion, erosion, sine wave wobbliness, zig- zagging and full supporting cast. Crack Art rivals the big-league art programs and leaves lesser ones floundering in its wake."

EXTRACT FROM ST FORMAT, JULY 1992, PAGE 31

Crack Art Version 1.00

"Ten PD Programs You Shouldn't Miss"

First Place: Crack Art

"A gorgeously wonderful and scrumptious art program. It does all the normal stuff and a heck of a lot more besides. Like what? Like splines (smooth curves automatically drawn between points you set), or block effects, or being able to wrap pictures around tubes and spheres, or fast and flexible zoom mode. You need 1 Mbyte to use the program and the instructions are in German, but don't let that deter you. The icons are in English, it's great fun to experiment with, and it's a great program, well designed and executed."

Review: STF 34 STF Rating: 94%

(For German readers, be it known that the Atari color monitor, not the monochrome, is standard in the United Kingdom.)

EXTRACT FROM ST-MAGAZINE, OCTOBER 1992, PAGES 120-121.

"Crack Art presents itself as a comprehensive color graphics program which leaves hardly any wish unfulfilled. ... Naturally, the many special functions are unlikely to be used without genuine study of the handbook, but that is simply the price for the extraordinary selection of graphics tools which hardly any other program offers. Whoever would like to be creative in color should not overlook this program."

EXTRACT FROM ST-COMPUTER, OCTOBER 1992, PAGE 150

Crack Art Version 1.30 TT+

PD Disk #550

With this program you have a powerful tool for creating and reworking graphics in Atari ST low resolution. An enumeration of the many functions would probably overflow this page (90 lines, 4 columns!). Only one thing need be said: Everything you need in a convenient drawing program is in there (even animation). Surprise yourself!

Suggestions

Should you have programmed something for Crack Art or have some kind of PD add-on which doesn't belong to the original package, we'd be glad to get a copy. We might even add it to the package.

We are always glad to receive suggestions (see T-TRIS). Inquiries about letting the program run in TT middle or low resolution are already in hand, but Crack Art runs only in ST low. What are we shutting out?

Crack Art Statistics

The source code for CRACK ART consists of:

- 15400 lines of GFA Basic (...and a few squashed ones)
- 164 principal assembler routines (compressed, about 1.5MBytes)

The graphics require about 164 Kbytes.

Users who are ready to pay the DM30 shareware fee for the 490KB (unpacked) main program pay practically nothing per byte. Drivers and games do not count. Therefore, smile and pay up! It goes without saying that you should first of all make a ten pack of backup copies [German beer comes in ten packs.]

Blowing our own horn...

The development history of Crack Art is entertaining for newcomers, though perhaps not for people who have used the program or know us.

From the authors (early 1990)

JAN 'JAYBEE' BORCHERS

It all started with a Holstein... No...and...yes... It all started with the old Atari 800. As Atari then introduced its new XL series, I traded for a 600XL, later for an 800XL, which after a short time was traded in toward a pool table. Suddenly in 1987 an Atari 1040 STF stood on my 2.65 meter [8'9"] wide computer table. An Atari 520 ST was added to enable faster programming and test runs. (Networking research has unfortunately remained just that.). The 520 ST was soon sold again and in the meantime the 1040 grew to have little in common with the purchased item (PC case with two drives, modem, switchable operating system, built-in video digitizer, sound sampler, MHz indicator and NMI interrupt. The stereo and the sound distorter were taken back out.) Had you noticed? I'm a hardware hacker. Next is my TT8 (crappy sound from the internal speaker).

Yes, the TT. In spite of "incompatible" (Hey Julian!) programming instructions, Crack Art runs (races) on it as well!!! Since we wrote most routines in pure assembler, the faster processor thanked us with insanely fast processing speeds. Only the Falcon gets a binary bellyache when we push Crack Art into its memory. (That was for adults only!)

Back to CRACK ART. Why the name? There is a diskette monitor by the name of CRACK, which, not yet full-fledged, was designed for hacking about in every data Babel one could find on a disk. After we put this project on ice and decided on a pixel-oriented drawing program to fulfill the same demands, we kind of adopted the name. To account for the artistic part, we tacked on the ART. The name was there; all we needed was the software.

[I've omitted a plea for someone to translate this document into English. Somehow it seemed redundant.]

Before the gentleman next to me parts my hair with the keyboard because I'm blithering on too long and too dumb, I'll turn the floor over to him.

We still speak, though:

```
hey_roy: move coffee,jay
          pea  please
          bsr  here
          lea  thanx(pc),a0
          .
          .
          .
          clr.w  -(sp)
          trap  #1                ; program end...
```

DETLEF "ROY" RÖTTGER

I had my first contact with computers when a classmate showed me his newest acquisition. I don't remember its name anymore. Anyway, he soon traded it toward an Atari 400 (Price at that time about DM 1250 [\$833 at today's rate of exchange]). With this computer, astounding things were possible, and my interest grew. Later I bought myself an Atari 800XL. Shortly before I started college, I blew my savings on an Atari 260ST, which I still have, along with the 800XL, by the way.

After experiencing various painting, drawing, and construction programs, we wanted to develop our own program, which would be superior to other programs in speed and especially in ease of use.

That brought Alex, at the time an Information Sciences student in Erlangen, to the careless statement that this project, too, would be broken off before it was finished (sadly, it has often happened). He has long abandoned that box of beer (and still has not gone back to it. Typical! And today, November 9, 1992, still not...).

CRACK ART grew and grew (In the meantime, the expression giant baby would be a shameless understatement. Have you ever imagined Arnold Schwarzenegger in a baby carriage? What is that supposed to be? Who was that? Jan, I AM writing..). It became a goal in itself as we implemented more and more new ideas. (That's kind of got out of hand and we're now turning more to add-on programs.)

Existence to everything - no program without its own picture format. The depackers and packers for the various graphic programs were so numerous that the idea for a pack algorithm of our own grew and grew. Thanks to some tricks, the packer compresses very well. In any case, I know of no graphics program that packs smaller than Crack Art. (The internal menu graphics were packed with an LZW packer of our own when Crack Art got too big to compile in 1 Mbyte. This packer works logically somewhat better!)

Since we have added one more format to the graphics format jungle, we've included the original source code for the packer and unpacker and a small demo listing so that they can be built into other programs. We have just recently got translations of these programs in C++ from Australia.

CRACK ART OVERVIEW

We have tried to make the handbook as thorough as possible, though it is easier to show the operation to someone than to describe it in hairsplitting detail (One picture says more than a thousand words). Should a function seem unbelievably complicated at the first reading, you are probably right, but in practice it will prove to be simple and useful. Therefore, you should try each function as you read about it, online as it were (assuming you have printed this out or own two computers).

With this program, you have a powerful tool for producing and modifying graphics in ST low resolution. We have had great fun with the development, and we would not like to be without the program. We believe that you will see that when you use the program and moreover that opportunities for creating useful programs are not yet gone.

We hope the explanations in this handbook are not too dry. We have strewn numerous small examples (and occasional wacky comments from Jan...) liberally in the text. These should clarify the operation of the program and lead the user through the tiny details.

If you're wondering why you can't get inside the program to see how the various functions work, it's because we've often used a chaos of little assembler routines paired with system graphics routines to achieve the total function, in whose result you're interested, not in the process. (Or, how do they do that?)

For those in a hurry, there is a list of keyboard commands for the elementary operations at the end of this handbook. These alone, however, will not clarify most functions. After that is an appendix with the HAVE YOU EVER SEEN...? details of the program.

On the Construction of the Menu



The various functions are represented on the menu by colored icons. You will immediately distinguish five groups:

- Icons for the elementary drawing functions (in shades of gray)
- Icons for block operations (multicolored)
- Icons for disk operations (disk icon)
- Icons for choice of screen (numbered in sequence)
- Icons for special functions (strange)

Functions are selected with the left mouse button. Drawing functions can also be selected via hot keys, (keyboard shortcuts) which are shown in red (e.g. 'D' -> DRAW). Some drawing functions allow specification of parameters after a double click. The hot keys for these are underlined. (Here it is, Marco!) The right mouse button toggles between the menu and the drawing screen(s).

Exit the program by clicking on the desktop icon in the lower right corner. If all drawing screens are empty, the program simply quits. Otherwise, it enumerates the occupied screens and gives you the opportunity to save or abandon them.

Time is easy to set after you have clicked on it. After a single click on the date, a calendar of the current month appears. You can change the date with SET. You can also change it by double clicking on it. The day of the week is automatically calculated. The displayed time changes only when you click on a menu selection. (Actually, we should not dwell on such trivialities. After all, who sits hours at a time in front of a paint program if he needs a watch?)

General Overview of Alert & Dialog Boxes.

ALERT BOXES

The alert boxes used in CRACK ART were entirely developed by us so there would be no discontinuities in the appearance of the menus. Their function is easily explained. All alert boxes contain up to ten lines of text and as many buttons as necessary in the bottom row. Default selections are ringed with red and can be selected with RETURN. Otherwise, make your choice with the left mouse button. All buttons have exit status.

DIALOG BOXES

The dialog boxes are somewhat more flexible in construction than the alert boxes. The first line of text names the function of the alert box, after which usually follow several lines denoting a function and buttons to activate them. The last line usually consists of the buttons 'OK' and 'CANCEL.' Currently active functions are ringed with red. Parameters can be changed any way you wish. They will take effect only after you leave the box by clicking on OK and the next time you call the dialog box, your last inputs will show as current, i.e. ringed in red. If you click on CANCEL, the old values remain.

FILE SELECT BOX

For a thorough explanation, see under Disk Operations. It is urgently recommended that you read this part of the manual carefully. Then you'll be spared such questions as "How much room is left on the disk?" or "Why can't I just load out of a horde of pictures and expect Crack Art to recognize the format automatically?" This, and much more, is offered by the file selector box.

SHAREWARE



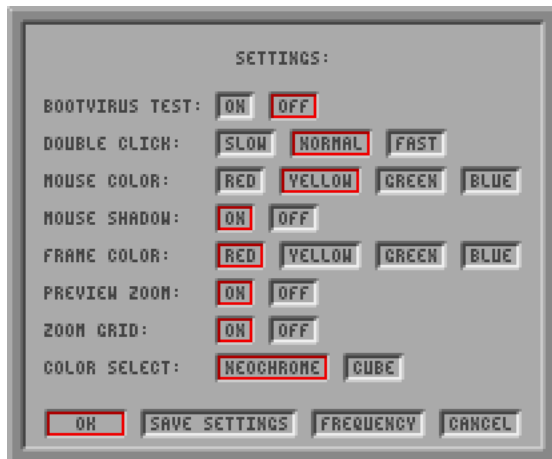
You should have at least read it once, even if most of it seems to be, er, fertilizer. Then think where you can get so much program for so little money (DM30, remember?). The five little games would be worth that!

Under REGISTER a dialog appears, into which you input the code which we send you after you send us the shareware donation. Type the five lines in exactly as they are given, and a file is created which turns off the limitations of the demo version. This copy is for you alone. Distribution of Crack Art with a registration file is forbidden! The name of the registered user appears, along with other information, in the info text at the beginning of the program. Support the shareware concept!

SETS



The very first button in the menu allows global parameters to be set.



Bootvirus Test turns the boot sector test of the boot drive, which Crack Art can carry out when you start it, on or off. If the boot drive is either A or B, drive A is checked. Except for Crack Art's own boot sector, every executable boot sector will be reported as a possible boot virus.

Double Click accepts the velocity setting of the mouse double click.

Mouse Color enables the selection of cursor color.

Mouse Shadow toggles mouse shadow on/off.

Frame Color allows you to choose the border color of selected icons. This is especially useful when you use a TV as a monitor and a particular color is most visible.

Preview Zoom toggles the real time loupe between the function keys and SHIFT+F-keys on and off.

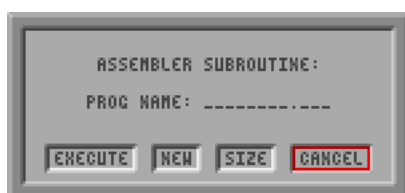
Zoom Grid determines whether you work with a pixel dividing raster in the loupe.

Color Select alters the appearance of the color selection when you double click on a color in the color menu (which see).

When **SAVE SETTINGS** is selected, these parameters are saved in a file called CRACKART.INF. The next time you load CRACK ART, they will be automatically set. In addition, driver buffer sizes, which are not shown in this dialog, are saved within the program.

A press of the **FREQUENCY** button causes the user to be asked if he really wants to change the display frequency. Switching from 50 to 60 Hz overdrives some monitors, especially TVs. Therefore, a special warning appears. (No effect on the TT.) When you save the configuration data, the frequency setting is also saved and used whenever you load the program again.

ASM



Should you somehow want a function we have not built in already, you can hang one of your own, up to 8192 bytes, here. From the stack, the routine will get the address of a structure, which is thoroughly explained under ACC. Be very careful not to exceed 32000 bytes per screen, or the program will probably crash. An example follows:

```
; -Example-----
      .TEXT ; Save registers before proceeding.
      movem.l d1-d7/a0-a6,-(sp)

;These addresses are on the stack and can be fetched like so.
;(Caution! When using this simple method in subroutines, pay
;attention, because the offsets are adjusted by the return
;address)
      move.l 60(sp),a0 ; Address of the transfer data structure
;Your routine starts here
```

```

        .
        .
; For example, inverting the current screen:

        move.l 4(a6),a0      ; Address of the current screen
        move.w #7999,d0     ; 8000 long words = 32000 Bytes
loop:    not.l  (a0)+        ; invert 1 long
        dbra   d0,loop      ; next long...

        .; And ends here
;Before returning, restore the registers
        movem.l (sp)+,d1-d7/a0-a6
        rts

        .DATA
        .EVEN

label_1: .DC.W ...
        ...
label_n: .DC.W ...

        .END;

```

The routine must be written completely PC (program counter) relative and, naturally cannot contain a Block Storage Segment (.BSS). Assemble as a .PRG. If you have not mastered Assembler, forget the whole thing here and now. If you are a C professional (I hate C! - Jan [Me too! - ed], you might try that. (Times change, and now we program almost exclusively in C, but not on the Atari.) If, contrary to expectations, you succeed in writing an add-in in C, we would be obliged if you would send us a copy of the source for public distribution. Additionally, you can use the Physbase as working storage. (That's really for the assembler vandals among us. But we love it!) On return from your routine, the menu is redrawn.

BLOCKS



The menu may give you the idea that we are crazy about blocks. To be honest, we carry out nearly all our picture manipulations with blocks. On the other hand, we create our pictures with the loupe. This is one reason why the loupe got special attention, but more on that later.

How, then, can one cut out the desired part of the screen? CRACK ART offers four (!!!!) possible ways to cut out a block. The ESC key always works. If you change your mind or find an error, just press ESC again or press the right mouse button and the previous block is still there. But, if you have already begun to mark the new block and then change your mind, you lose the previous block.

1. Cutting out a rectangular block

After you press the ESC key, a screen-sized cross hair appears. The current X and Y coordinates are shown in the lower left corner. With the mouse, move the cross hair to one corner of your block. It does not matter which corner. Press the left mouse button and hold it down. The cross hair disappears, and a rectangle appears. Moving the mouse, draw a frame around the block you want. When you release the left button, the block is captured, and all functions are directed at the block. Depending on which function was previously active, either a small, hollow cross hair or the block itself will follow the mouse motion. By way of explanation, all

functions which do drawing operations on an area will use the block and its associated background pattern as a fill pattern. If you want to use the block as a pure fill pattern, a fully filled background pattern must be active. (Background patterns are set in the color menu. See there.) If you want to see your block, turn the DRAW function on by typing D. This guarantees that your block is tied to the mouse. You can draw as usual but marked with the block. With the B key, you toggle between Block and Brush.

2. Cutting out a round block

If you want to cut circular or elliptical blocks (Yes, that works, too.) press CONTROL+ESC. The large cross hair appears again, with which you mark the center of your ellipse. Press and hold the left mouse button while you pull an ellipse around the desired area. See CIRCLE, above, for a review. When you release the mouse button, the round (elliptical) block is cut out. From here on, everything works as it did for rectangular blocks, only the block mode is OBJECT. More on that at the end.

3. Cutting a block with a lasso

If rectangular or round blocks are not enough, CRACK ART offers the possibility of cutting out blocks of arbitrary shape. Press SHIFT+ESC. A small cross hair, like those used by the drawing functions, appears. Move the cross hair to the start point of your cut and click (do not hold) the left mouse button to mark the point. With more clicks of the left button define the desired region. Or simply hold down the left mouse button and draw around your desired block. Some programs allow maybe only 27 or 1000 corners on your polygon. You do not have to worry about such piddly numbers with CRACK ART. You will probably never need its maximum 10,000 points. Click the right button when you are done. Now, with the left mouse button, you can choose either what is inside or what is outside the loop as your block. From here on, everything works as described for rectangles and round blocks. TIP: If you want a block in the form of wild lines, scribble all over the screen, then cut off a tiny loop at the end. Only this part will be filled. Depending on the density of lasso lines, the resulting block is more or less recognizable. The block mode is OBJECT.

When selecting a block by one of the three foregoing methods, if you hold down the ALT key, the spot occupied by the block will be erased when you cut the block out. You will have a hole in the picture, which you can restore with the UNDO key.

4. Cutting out a block via a fill function

If you want to cut out a complicated looking (but connected) block, press ALT+ESC and click on the screen region which you want to take as a block. Pixels of color 0 (background color) will be interpreted as borders of the block. The object within the border will be taken as a block. The block mode is OBJECT.

What do the various block modes have to say?

In the menu you will see in the top row a button with an artist's paintbrush on it. To its right is a button with a highlighted block. You can toggle back and forth between them with the

mouse or the B key (radio buttons). Farther to the right are the buttons for setting the block mode and the block boundaries on the screen.

XRAY

In this mode, all parts of the block having the background color (color 0 as Key Color) are transparent.

OBJECT

This block mode is something quite special. It is produced by cutting with the lasso or the ellipse, through which the block mask is determined. All regions within the mask are opaque. If this mode is inactive, it can be turned on via the menu and produces an object mask. (See CUT and OBJECT. Yes, the same button is also used to produce an object mask.)

BLOCKED

In this block mode, all regions of the block are opaque (rectangles).

More about block modes:

You can switch back and forth between XRAY and BLOCKED without loss of information. You can use the menu buttons or the left and right parentheses on the keyboard. If you shift from OBJECT to another block mode, you lose the old object mask.

For example, take a block containing an unfilled circle. If you cut this block out in circular form, probably not exactly centered on the unfilled circle, you will have an Object Block with a mask which renders the inner area opaque, but which follows the contour well. If you switch to XRAY, the circle is transparent inside and out, but in BLOCKED mode, the circle is opaque inside and out, with a rectangular mask surrounding. Now, when you apply the OBJECT function to this block, you get a block that is opaque inside and transparent outside, with the circle perfectly bordering the mask. [Remember 'way back in the introduction when the authors suggested trying the various functions as you go along? This is a good place to do so.]

BORDERING THE BLOCK WITH THE SCREEN EDGES

With the buttons FREE-MOVE and EDGE-STOP, or the '/' and '*' keys, you can toggle back and forth between the two kinds of borders. EDGE-STOP means that the block cannot pass beyond the screen boundaries, while FREE-MOVE lets the block cross halfway over the boundary, that is till the middle of the block reaches the boundary.

THE BRUSH FUNCTIONS



All the Brush-functions are explained here. The icon of the function is displayed on the top right. The icons after it indicate the modes the function is available to use in;



= Brush

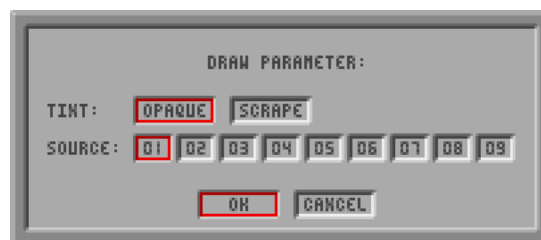


= Block

DRAW



Freehand drawing with the mouse button down.



When Opaque is selected, you cannot see through the drawn lines. With Scrape you scratch through the background of the screen selected via Source. This function is especially handy for trimming blocks whose edges were not neatly cut out.

Limitations: You cannot draw polylines with Opaque in block mode.

Hint: Use slow mouse movements. Scrape works only pointwise and not with Block. Scrape large areas with BOX (which see).

POINT



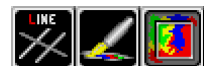
Set single points.

Point, in conjunction with the SHIFT and CONTROL keys, turns on special functions.

SHIFT: The block is copied behind the current picture.

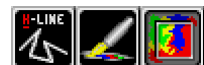
CONTROL: The block is copied onto the current picture and the part of the picture that was covered up becomes the new block. With this, arbitrarily complicated blocks (especially in OBJECT mode) can be brought onto a picture. Try it once!

LINE



Draw lines by specifying the end points.

K-LINE



Draw arbitrarily long chains of connected straight lines or regular polygons. End a line chain with the right mouse button. On completion of a polygon, the mouse is set in the center so that you can draw concentric polygons without having to mark the center point.



Limitations: Blocks cannot be used with regular polygons.

FRAME

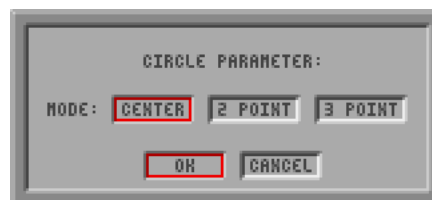


Draw rectangles by marking two corners with the mouse.

CIRCLE



Draw circles and ellipses. On completion of a circle or ellipse, the mouse is set at the center so you can draw concentric circles or ellipses without having to locate the center point.



In center mode, circles and ellipses are specified with the center point and the radius or two semiaxes. To get true circles, hold down the ALTERNATE key while drawing.

In 2-Point mode, you mark two points which are taken as the ends of a diameter and the circle is drawn between them. In 3-Point mode, you specify three points (not in a straight line) and a circle is fit to them.

Limitations: CIRCLE works in Center mode only with the block.

ERASER



Brush: Mark the size of your eraser with a rectangle. With it, you can wipe the screen clean, except for the frame of the rectangle itself. Resize the rectangle after pressing the right mouse button.

Block: The block is your eraser. The form of the block comes from the block mode (which see).

AIRBRUSH



Spray can



Before spraying, set the spray size with a circle.

Limitations: Works only with Brush.

RAVS



Draw lines with a common start point.

POLYGON



Draw filled polygons, regular or no. Use the right mouse button to complete irregular polygons.

Brush: The polygon will be filled with the current background pattern.

Block: The polygon will be filled with the background associated with the block.



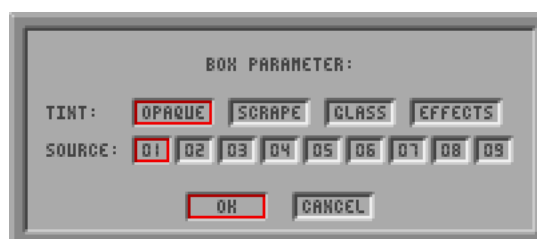
Limitations: Not more than 128 sides, please.

BOX



Draw a filled rectangle or carry out a special function inside a rectangular region. (One of the most powerful functions!)

Set two corner points of the rectangle with the mouse.



Brush: The rectangle will be filled with the current background pattern.

Block: The rectangle will be filled with the background pattern associated with the block. (Try a checkerboard pattern and a multi-colored block.)

Scrape: A rectangular region will be 'scraped' out of the screen specified by Source and the colors will be adjusted simultaneously.

Glass: The current color will be applied transparently over the rectangle. (Good for multiple levels of a color.)

Effects: An effect routine will be applied over the rectangle. The effects are explained in EFFECTS. They correspond to the number in Source:

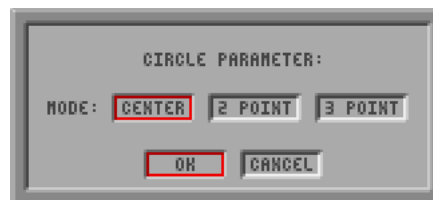
1 = ANTI 4	5 = EROSION
2 = ANTI 8	6 = MEDIAN -
3 = ROUND	7 = MEDIAN
4 = CONTRAST	8 = MEDIAN +
9 = DILATATION	

Limitations: With a block, only opaque mode is available.

DISC



Draw filled circles or ellipses.



Same as CIRCLE, above. In brush mode, the current background is the fill pattern, while block mode uses the pattern associated with the block.

TEXT



Add lines of text with either standard or a user-defined font. Specify the standard font's parameters in the Color menu (which see).

Crack Art makes no distinction between fonts and pictures. Fonts are pictures in which the individual characters, beginning at the upper left with an arbitrary character, are arranged in rasters per the ASCII sequence.

The maximum size of a character is 96*96 pixels in 16 colors. If, as a result of editing a font, a character no longer fits on the current line, it must be moved completely to the next line. In this way, Crack Art handles organized characters properly.

Specify size by double clicking on the TEXT button with the option SET. Then you can specify width, height, the first character of the font, the spacing between characters, the width of the space character, and the source screen on which the font is found.

These parameters can be saved in a DEF file with the same name as the font picture via SAVE SETTINGS. At load time, the source screen will be adjusted to the current screen. If a CA1, IC1, or PI1 is loaded, Crack Art looks for a DEF file which identifies the picture as a font picture. If it finds one, it takes the settings as text parameters and switches them into the color menu under EXTRA and PROP so that you can work with the font immediately.

Fonts in E24 and FNT format, on the other hand, are loaded into the parameter dialog of the TEXT button and expanded onto the current screen.

- Width:** width of the characters
- Height:** height of the characters
- First Char:** ASCII value of the first character in the font
- Space:** width of the space character
- Distance:** space between two characters
- Screen:** screen number of the user-defined character set



SAVE PARAMETERS:

A file with the name of the current screen and the extension DEF will be written with the current parameters. When a picture in the format CA1, IC1, PI1, or PC1 is loaded, a matching DEF file will be sought, and if found, read. A renewed setting will be left off in the future.

LOAD E24 FONT:

To load SIGNUM! editor fonts: (Caution! The font will be expanded onto the current screen.) The characters are 16x24 pixels and are always in color #15. (Can be changed with EXCHANGE in the color menu.) These values will be automatically set. DISTANCE is set to 1. In the color menu, EXTRA and PROP are chosen in order to be able to work with the fonts immediately.



To load editor FNT fonts (TEMPUS, EDIMAX, DEGAS...) (Caution! The font will be expanded onto the current screen.) Up to 8x16 pixels, the above holds.

Comments: ESC erases to the beginning of the text line. If the line is empty, text mode is abandoned, and DRAW turned on.

Limitations: Multiline text is not editable.

OUTLINE



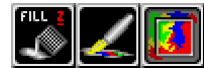
Put a colored border around an area. Select the area and/or the color with the mouse.

- Form:** Mark the corners of the border.
- Part:** Put a border around only this area or around all areas of the same color on the screen.
- Mode:** Put a border around only the selected color or bring all neighboring colors into the area.
- Type:** The bounded area remains or is wiped out.



Limitations: Doesn't work on the background color.

FILL



Fill areas (Flood Fill).



In opaque mode, filling works as follows:

Brush: The current background pattern is the fill pattern.

Block: The background pattern connected to the block is used.

In dither mode, you can fill the area with a color sequence. After you click on the area to fill, a cursor with the name VEC appears. Use it as if drawing a LINE which shows the direction of the color sequence. Then you will see a color palette. Pick the starting color with the upper arrow and the ending color with the lower arrow. Unlike the case with DITHER, only the start and end colors are used here; any in between are ignored. Click the right mouse button to start the fill process. (Filling takes the most time in this function. Dithering proceeds in real time on the TT!)

With Matrix you install a matrix to be used for the dithering. The names of the matrices reflect the sizes of the points they generate. HUGE generates big blobs!

SMEAR



Wiping of pixels.



Limitations: Works only in Brush mode.

MIRROR



Reflection about axes of symmetry.



Horizontal: Reflection about the horizontal axis of symmetry.

Vertical: Reflection about the vertical axis of symmetry.

Both: Simultaneous reflection about both axes

Limitations: Doesn't work with

ERASER, AIRBRUSH, TEXT, OUTLINE, FILL, SMEAR, SPLINE

Should one of these functions be active when MIRROR is selected, the program shifts automatically to DRAW. If you select one of these functions while in MIRROR, you will be switched out. The fill pattern of area-related functions like POLYGON, BOX, and DISC is not reflected.

SPLINE



Drawing spline curves with two different algorithms and arcs between two points.



B-Spline: Spline that is tangent to a polyline

Smooth: Spline that passes through the corner points of a polyline

Curve: Arc between two endpoints. Intersection (below) is ignored.

Intersect.: Number of support points between two end points.

Limitations: Doesn't work with Block.

The Working Screens

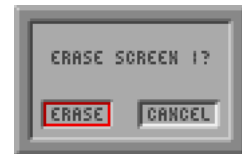


In the lower left corner of the main menu, you will see nine numbered buttons, each with two light diodes. They represent the working screens. Depending on memory size, the available screens will be marked with blue LEDs on startup. (The blue LEDs have meaning only for animation. See SCREENS.) The current working screen is marked with a red LED. Switch screens with the mouse or by typing the corresponding number. The right mouse button toggles you between the current screen and the main menu. The ESC key will get you back to the current screen from the main menu, but the reverse DOES NOT work. (In the current screen, the ESC key cuts out a block.)

Functions with the Working Screens

SCREEN ERASE:

Holding the left mouse button down, drag the desired screen icon into the trash can. For safety, the program asks whether you really want to erase screen #n.



COPYING, SWAPPING, FLOODING:

Drag one icon onto a second. A dialog box appears, allowing different modes and, if it makes sense, recolor to match the current palette.

(Warning! If you flow with ALL or IMAGE, you overwrite all screens with the current one!)



ADD TO ANIMATION LIST:

A double click on the working screen icon turns the blue LED on or off (if the screen is available). The screens with the blue lights on are in the animation list; the others are not. (See SCREENS.) By the way, if you hold the mouse button down on an unavailable icon, the current one will be grabbed!

SHOW



Next to the nine buttons is a similar one, labeled SHOW. With this one it is possible to get an overview of all active screens.

A double click on this button allows the setting of parameters:



(TT-View works only on the TT with at least four screens, therefore not in the unregistered demo version.)

With Black/White, all occupied screens will be drawn in little gray-scale pictures. That takes a certain amount of time. After that, little frames are drawn around them all simultaneously. Under each picture is its name or a blue DISUSED. Empty screens are marked with a graffiti EMPTY SCREEN.

With Color, all nine (or as many as available) screens will be drawn in reduced size using the color palette of the current screen. Thereby, it is possible to draw pictures larger than one screen. In order to match the edges of the component pictures, each can be "rolled out" like piecrust. Use the cursor keys and the pictures will be copied in the corresponding direction by one half the screen width or height. (Only the available screens, naturally.) This makes sense only for 1, 2, 3, 4, 6, and nine pictures. (You will see why when you try it.) The working screens will then show the seam lines and you can doctor them to match.

Finally, all pictures can be restored to their original positions (use the same procedure in reverse).

With TT-View, screens 1-4 are shown in middle TT resolution with the current color palette. (Be sure to have saved your work before trying this one.)

THE COLOR MENU

The color menu allows access to some very powerful functions which are suitable not only for color manipulation but also for direct picture and block manipulation. As one of the primary tools in CRACK ART, it should not be underestimated.



Because CRACK ART supports up to nine work screens with individual color palettes, it is pointless to set colors from the main menu. Thus, the color menu is reachable only from the work screens. To get it, press the SPACE bar, or, with the mouse cursor on the extreme edge of the screen, press the right mouse button. A pop-up menu will appear, with the 16 current colors displayed side by side. Above and below the color register are arrows. The upper arrow marks the current drawing color and the lower one points to the so-called EXCHANGE color, which we will cover below. Choose the drawing color by clicking on the register, by left clicking on a color outside the menu (the color under the cursor will be selected) or by pressing the right or left cursor keys.

On the right side of the menu are sliders for the red, blue, and green components of the drawing color. Adjust them by clicking on the upper or lower arrows on the sliders. Additionally, by double clicking on a color register, you can activate a raster interrupt color selection, which allows setting the register in Neochrome style. Besides the Neochrome variant, the CUBE mode (selectable under SETTINGS) is available.

By clicking and holding on the striped button on the left side, you can push the menu around the screen and see the underlying details.

Also, in the menu are two larger rectangular areas which show the current fill pattern and the current character set. Clicking on the fill pattern box calls up another menu containing 80 readymade fill patterns. The current one is ringed. Select with the left mouse button. The new fill pattern will be displayed in the color menu. The 80 fill patterns cannot be edited. The possible fill patterns in CRACK ART are far too numerous to be limited to 16x16 pixel units. Since you can use any block as a fill pattern, you do not need to diddle with the built-in patterns. You can get around this easily since any block can be used as a fill pattern.

The font display box always contains the 'A' of the current font. For user-defined fonts, this is a useful check for the right sizes (set with TEXT-SETS). Font attributes are set with the SYSTEM, EXTRA, and PROP boxes. The current settings are displayed in reverse video. Change the size of the system font with the buttons marked with plus and minus signs. BOLD and LIGHT are self-explanatory. These buttons have no effect on user-defined fonts.

If you choose the EXTRA option, the parameters you set by double clicking on the TEXT icon in the main menu are taken on here (SETTINGS). If you then turn on PROP, the user-defined fonts are displayed as proportional fonts.

Options for manipulating the color palette

PREDEFINED COLOR PALETTES:

The function keys are loaded with 20 predefined color palettes. Palette F1 corresponds to the original ST palette and palette F2 to the preset CRACK ART palette. Palettes F11 through F20 are reached via SHIFT + F1 through F10. Many of these palettes are useful for gray scale pictures or digitalized or scanned graphics. They are formed from color sequences from black to very bright tones.

COPYING A COLOR TO ANOTHER REGISTER

If you hold the left button down on a color register, you can drag it to another one. When you release the button, the copying is complete.

SWAPPING TWO COLOR REGISTERS

Proceed as above, but before releasing the mouse button, press the ALT key.

PRODUCING A COLOR SEQUENCE

If you want a uniform or proportional color sequence between two registers, click on one, then press ALT and click on the other. All color registers between them will be filled according to their position in the color palette with a segment of the calculated color sequence.

DETERMINING THE COLOR OF A POINT ON THE SCREEN

If you move the mouse pointer outside the menu boundaries, the arrow becomes a cross hair. Clicking the left button causes the color of the point under the cross hair to become the current drawing color.

DIRECT COLOR SELECTION

If you double click on one of the color registers or outside the color menu, a raster of more than 16 colors appears. Control the construction of the raster through the COLORS option in SETTINGS. NEOCHROME gives you exactly the Neochrome color arrangement. CUBE gives you a stacked arrangement. (The faces of an RGB cube with sides of length eight.) Pick your color with the tiny frame [the cursor shape]. When you click the left button, the corresponding color value in the current color register will be brought over. The right mouse button ends the process.

Special Functions:

SORT: Sorts the colors according to brightness. The pixels of the picture and the block will be changed accordingly.

INV: Inverts the color palette. If you hold the ALT key down, a photo negative palette is created.

INV+EX: Analogous to INV, but changes the picture, too.

STATS: Displays a tabular overview of the picture points on the screen or in the block, depending on whether BRUSH or BLOCK was selected in the main menu.

EXCHANGE: Exchanges the pixel colors according to the exchange rules set by the arrows below the color register. If, for example, you choose color #1 and its exchange arrow points to color #2, color #2 will replace color #1 everywhere in the picture or block. You can do several colors simultaneously. In this way you can correct colors after swapping registers. **IMPORTANT!** If you selected BLOCK in the color menu, exchanges work only on blocks. For this reason, EXCHANGE is a very powerful block function.

With the following combinations (a key and the left mouse button), no pixel exchange will occur: (e.g. hold the ALT key down while pressing the left mouse button.)

ALT+EXCHANGE directs all exchanges back onto themselves.

SHIFT+EXCHANGE rotates all exchange colors one position to the left.

CONTROL+EXCHANGE rotates all exchange colors one position to the right.

SET PLANES: Enables in-blending or out-blending of single bitplanes via the buttons 0 through 3. The current arrangement of planes can be taken over with a double click on SET PLANES. Exit by clicking the right mouse button.

BRIGHT+/-: Brightens or darkens the entire color palette.

RGB+/-: Brightens or darkens the primary colors of the palette.

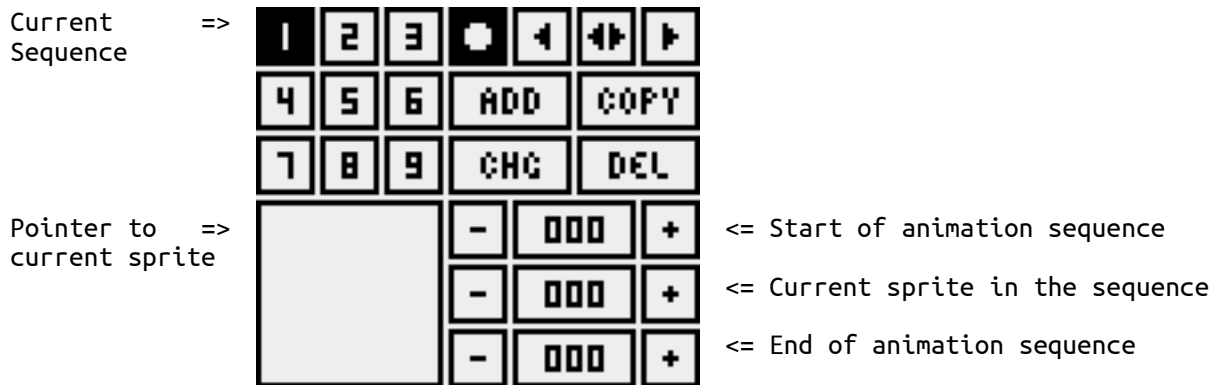
Most functions which alter the color palette can be undone to the previous condition with the UNDO key.

Keys 1 thru 9 switch among the work screens while the color menu is displayed.

Leave the color menu with the SPACE bar or the right mouse button.

Sprite Animation

The sprite menu is turned on and off with the button marked with a circle, next to the fill pattern display. Only when the menu is turned on are the buttons on the right side selectable and the sprite indicators activated.



ADD - INTRODUCING SPRITES INTO THE ANIMATION SEQUENCE

When you click on ADD, the color menu goes away and a crosshair appears, just as in the cutting out of blocks. Mark a (rectangular) section of the screen with the crosshairs. The coordinates of its corners will be copied into a sprite buffer and added to the sequence. If you hold the ALT key down when you click on ADD, the area you select will be marked by means of a fill function when you cut it out. This mode is marked by a special mouse cursor. IMPORTANT: The current drawing color will be taken as the boundary, that is, you can click on one of the rectangular frames around a sprite (of this color) and the region inside will be interpreted as a sprite region. If you hold down the CONTROL key instead of the ALT key, no new sprite will be created. Rather the sprite from the One-Sprite Buffer is introduced into the sequence.

COPY - COPY THE CURRENT SPRITE INTO THE ONE-SPRITE BUFFER

The current sprite is copied into the one-sprite buffer and can from there with CONTROL+'ADD' or CONTROL+'CHG' be either inserted into, or replaced in, the animation sequence.

DEL - ERASE THE CURRENT SPRITE

The coordinates of the current sprite are deleted from the sprite sequence and the coordinates of the following sprites are moved forward to fill the gap.

PLAYING THE SPRITE SEQUENCE

The sprite sequence marked with numbers 1 through 9 can be played with the buttons ◀, ◀▶, and ▶. The sequence starts with the designated start sprite and ends with the designated end sprite. The animation direction is backwards ◀, ping-pong ◀▶, or forwards ▶. The color menu disappears, and the animated sprite hangs centered on the mouse. Vary the animation speed by holding the left mouse button down and moving the mouse leftward (faster) or rightward (slower). The speed can be seen directly as the mouse is moved. The sprite position stays still during speed adjustments. The color menu comes back with a press of the right mouse button. Exit the color menu with SPACE or another press of the right mouse button.

BLOCK OPERATIONS



X-RAY



Function: The object mode will be set to X-RAY.

Description: The function creates a mask which is transparent to all parts of the picture containing no graphic (Color 0). The resulting block mode is X-RAY.

OBJECT



Function: Builds an OBJECT mask for the current block.

Description: The function creates a mask which is transparent to all picture parts outside the block which do not have a graphic. Parts inside the block, on the other hand, do not show through. Try it with a circle. Block mode afterward is OBJECT.

BLOCKED



Function: Sets object mode to BLOCKED.

Description: The function builds a rectangular mask in the size of the block. The entire block region is opaque. Block mode stays BLOCKED afterward.

FREE MOVE



Function: Allows movement of the block till the center of the block reaches a screen boundary.

EDGE STOP



Function: No part of the block may pass the screen boundary.

SIZE



Function: Changes the size of the current block.

Description: The current block is identified in the upper left corner. A frame marks the current size. The X and Y dimensions of the frame are shown in the lower left corner. Stretch the frame to the desired size with the mouse and left button. UNDO restores the previous size. HELP invokes interpolation after the size adjustment. This filters out some of the unattractive patterns that occur with simple resizing routines. The quality of the result is especially good in very small sizes.

Holding down the...

ALT key allows only the vertical size to change;
SHIFT key allows only the horizontal size to change;
CONTROL key holds the aspect ratio constant.
Right mouse button accepts the new block and takes you back to the menu.

Block mode afterward is XRAY.

ROTATE



Function: Rotates a block in angular steps.

Description: The current block is displayed in the middle of the screen. A frame marks the rotation position and a line from the center of the frame to the mouse position helps determine the angle, which is displayed in the lower left corner. Turn the frame to rotate the block.

Pressing the...

UNDO key sets block and frame back to zero rotation;
Left mouse button carries out the rotation;
HELP key causes non-aliasing rotation, calculating fractional degrees;
BACKSPACE key causes modulo rotation and tiles the entire screen with the rotated block. Modulo means the pattern repeats, here according to the width and height of the block.
Right mouse button accepts the new block and takes you back to the menu, with an OBJECT block mode, while the BACKSPACE key takes you back with an XRAY block mode.

Tip: Use BACKSPACE in the workscreen to rotate the block 90 degrees clockwise. (See: Keyboard Layout)

H-SKEW



Function: Horizontal shearing of the block.

Description: The current block is displayed in the middle of the screen, surrounded by a parallelogram showing the current skew position. Change the parallelogram to distort the block.

Pressing the...

UNDO key sets the block and frame back the way they were;
Left mouse button carries out the operation;
Right mouse button accepts the new block and takes you back to the menu with an XRAY block mode.

V-SKEW



Function: Vertical shearing of the block.

Description: The current block is displayed in the middle of the screen, surrounded by a parallelogram showing the current skew position. Change the parallelogram to distort the block.

Pressing the...

UNDO key sets the block and frame back the way they were;
Left mouse button carries out the operation;
Right mouse button accepts the new block and takes you back to the menu with an XRAY block mode.

H-PERSP



Function: Horizontal trapezoidal warping of the block.

Description: The current working screen is displayed as background. A trapezoid marks the current position and form of the block. Change the trapezoid to determine the future appearance of the block. When you press the left mouse button, the cursor jumps to the next corner of the trapezoid. As long as you hold the button down, you can drag the corner anywhere you like. Repeat with the other corners till you have the shape you want.

Pressing the...

ALT+L mouse button carries out the operation;
Right mouse button takes you back to the menu with an XRAY block mode.

V-PERSP



Function: Vertical trapezoidal warping of the block.

Description: The current working screen is displayed as background. A trapezoid marks the current position and form of the block. As with H-PERSP, change the trapezoid to determine the future appearance of the block.

Pressing the...

ALT+L mouse button carries out the operation;
Right mouse button takes you back to the menu with an XRAY block mode.

CURL



Function: Radial skewing of a block.

Description: This function strongly resembles the BULGE routine in speed and operation. With the mouse, you specify two radii, a starting angle, and an ending angle. The block will be projected onto the specified region, whose maximum size is a circle of diameter 200 pixels.

The right mouse button stops the operation; block mode afterward is OBJECT.

Peculiarity: No UNDO is possible.

BULGE



Function: Spherical distortion of a block.

Description: This block function is one of the most successful routines in the entire program in terms of speed and outcome. First, a rectangle appears. In its upper half is shown a mirror symmetric filled curve. (The size depends on the block. Try some big blocks (No, not pictures of motors!) first.) When you press the right mouse button, the block transformation is carried out with this function. On completion, the block looks as if it had been stretched over a sphere.

If you would like to control the distortion function, you can alter the curve by holding the left mouse button down. The height of the function is a measure of the stretching of the block at this point (on a circular path). The function is not followed exactly, but generally acts like a lens placed very close. Experiment with middle sized to large blocks. The largest allowable ending block is a circle of diameter 200 pixels. Try to generate curves in the form of a normal distribution. (If you have trouble getting started, there is a picture on every new 10 Mark note.) [We Yankees will have to remember the bell curve.] The results will usually fit almost exactly back into the original picture. If you had a colorful geometrical object there, you would see after the transformation how [the artist? (Yes. But not with our small program, I think.)] Vasarely creates his works.

Since the ending block mode is OBJECT it can happen that circular blocks sometimes have a disturbing edge. Switch the block mode to XRAY or BLOCKED and generate a new mask with OBJECT.

Peculiarity: No UNDO is possible.

TUBE



Function: Pull the block around a tube.

Description: With the mouse, a line drawing of the tube can be altered in radius and viewing angle.

The function is activated with the left mouse button. The tube is built from back to front. Thereby, the rearward parts of the block, which would be visible from inside the tube, are darkened in order to give the effect of depth. Block mode at end is XRAY.

Peculiarity: No UNDO is possible.

FIT-IN



Function: Horizontal fitting of a block into an (almost) freely definable form.

Description: The current screen is the background. Specify a start point with the mouse cursor or abort with the right mouse button. You can specify up to 9,999 more in defining your form (identical to cutting out with the lasso). Click the right mouse button to show completion of the border, then click the left button either inside or outside to show which is to be used as the form. (Normally inside; outside does not make much sense.) Before fitting begins, the form is made "horizontally convex." That is, the leftmost and rightmost points on each row of pixels are linked together. Then the block is squished into the resulting form. If

you want to a form that is "vertically convex" rotate the block first, do the FIT-IN, and cut out to turn back. Ending block mode is OBJECT.

Peculiarity: No UNDO possible.

DITHER



Function: Five routines for producing a gentle color transition.

Description: IMPORTANT: The appearance of the resulting block depends on the block mode!



V-GRADE: Vertical dithering. A color sequence from top to bottom is produced.

H-GRADE: Horizontal dithering. A left to right color sequence is produced.

S-GRADE: Dithering according to the arithmetic average of the distance between the center and the pixel's coordinate. The color sequence proceeds linearly from a center outward.

M-GRADE: Dithering according to the root mean square of the distance between the center and the pixel's coordinate. The color sequence proceeds quadratically from a center outward.

R-GRADE: Radial dithering, using the Euclidean norm. The color sequence proceeds circularly outward from a center.

After selection of the desired method, a partial color menu appears: the 16 color registers, each with an arrow above and below. The arrows can be moved with the left mouse button. They cannot both point to the same color. The upper arrow points to the starting color - top side, left side, or center. The lower arrow points to the ending color - bottom, right side, or outer rim. Press the right mouse button to get things started.

V- and H-GRADE run in real time and start immediately. For the others, the block is indicated, and its size is shown with a rectangle. With the left mouse button, pick a dithering center inside the rectangle. The mouse cursor disappears, and the function is carried out in the background. (That takes a little while. For R-GRADE, up to 64,000 square roots must be calculated! Do not be afraid, I use integer arithmetic.)

By the way, the dither matrices are adjusted according to the size of the block. If the block is too small to process with the specified number of colors, S-, M-, and R-Grade do nothing. V- and H-Grade will return the pure colors in lines. Each routine has three dither matrix sizes available. That gives up to 64 steps between two colors!

The actual dithering is carried out differently according to the block mode. If the mode is BLOCKED, you get a dithered rectangle. For X-RAY, the color sequence is visible only where there is a graphic. That is, an empty block still looks empty and will therefore be shrunk to 1*1 pixels!

Try R-GRADE with a predefined palette with small color steps on a circular block in X-RAY mode.

CUT



Function: Minimize X and Y spread of the block.

Description: The block is trimmed by removal of empty edges.

Limitations: Doesn't work with block mode OBJECT. This function is on the HELP key in the working screen.

Other block functions in the working screen

BACKSPACE-KEY

Function: Turns block 90 degrees clockwise.

Description: The current block is rotated 90 degrees clockwise. This is quicker than calling up the menu and specifying 270 degrees (counterclockwise). The block doesn't rotate about its center point, but rather so that its left edge ends up on top. Pixels with X coordinate greater than 199 will be clipped by the lower screen edge and lost. If you want to preserve the right edge, flip the block with the cursor keys, then turn it.

RETURN-KEY

Function: Sets the block to the current color palette.

Description: Should the block come from a work screen with another color palette, you can try to reset the colors to conform to the current palette. The function tries to maintain as much as possible of the original colors and brightnesses.

CURSOR-KEYS

Function: Block flipping.

Description: The block and mask are flipped in the direction indicated by the cursor key.

CLR/HOME-KEY

Function: Block centering

Description: If it is visible, the block is centered on the screen. If you hold the CONTROL key down as well, the block is centered only horizontally.

THE MAGNIFYING GLASS OR ZOOM

In the top row of the main menu is a button called SETS. With this button, it is also possible to specify parameters for the magnifying glass. Clicking on the button brings up a dialog box on which the parameters Preview Zoom and Zoom Grid can be turned on or off. Preview Zoom

means that when you choose a segment to enlarge, a real-time magnifying glass is displayed. The portion of the picture under the magnifying glass is enlarged fourfold. The 12x12 magnifying glass (F10) enlarges 16-fold. By the way, it is possible to traverse the edges of the picture with this magnifying glass, unlike a well-known monochrome drawing program whose magnifying glass stops at the edges. The real time magnifying glass works in all 10 enlargements. If you turn the Preview option off, you get a crosshatched region marking the region to enlarge. Both options can also be reached with the function keys or the shifted function keys.

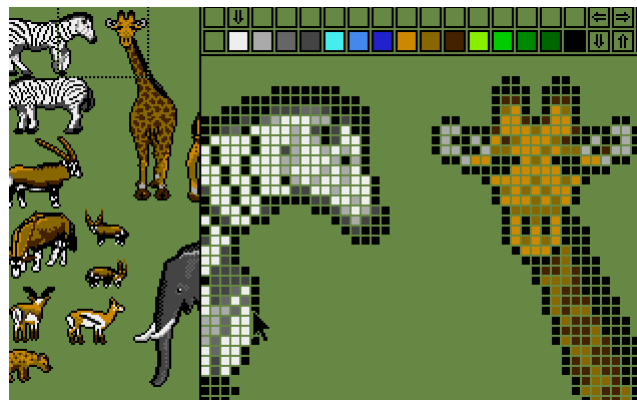
With the Zoom Grid option, the individual pixels are marked by a grid in the background color. (Since the background color is normally black, the display gets noticeably darker at small enlargements.) They are a little easier to distinguish this way. If you turn the option off, no grid is drawn.

How Can I Call Up the Magnifying Glass?

If you are in the working screen, you call up the ten enlargement levels with the function keys. Enlargements run from 3*3 (F1) to 12*12 (F10). Depending on the parameter values specified, the real time magnifying glass is on the function keys and the frame is on the shifted function keys (Preview Zoom On) or vice versa (Preview Zoom Off). Both options are always available. Once you have chosen which preview mode you are in, you can change enlargements without the SHIFT key. The right mouse button puts you back in normal drawing mode.

Choose the picture segment you want to enlarge and press the left mouse button to display the enlargement. You find yourself in the actual magnifying glass.

On the Construction of the Magnifying Glass



In the left part of the screen you will see a crosshatched, rectangular section of the working screen, which marks the enlarged area. The current color palette is above, with the current color marked with an arrow. Next to the color menu on the right are four arrows. By left clicking on one of them, you can cause the working area of the magnifying glass to move in that direction by 50% of the extent of the magnifying glass in that direction. The cursor keys function the same way. At the top right of the screen is a narrow box with which you can toggle the Zoom Grid without leaving the magnifying glass. The actual magnifying glass is at the lower right.

On the Functions in the Magnifying Glass

The magnification can be changed at any time by pressing a function key. The center always stays in the magnified area. You can pick a new center by left clicking in the original picture. The magnifying glass and the little frame will follow. Abandon the magnifying glass by right clicking outside the magnified area. (That is in the color palette or the original sized part of the screen!)

You can change the current color by left clicking in the color palette or by right clicking in the magnified area. The second method allows the current color of magnified sections to be changed quickly without constantly having to run back to the color palette. (By the way, this became a standard feature in newer paint programs!!) As fast as you left click on a pixel in the magnified view, the original is also changed. You can change it back with the UNDO key and change it back again with a second press of the UNDO key.

You can also cut out rectangular blocks from inside the magnifying glass. Press the ESCAPE key and a cross hair appears. Mark a corner of the block with it. The procedure then is the same as the normal cutting of a rectangular block, except the frame is doubled. If you hold down the ALT key when you release the mouse button, the block will be erased from the magnifying glass. The UNDO key restores it. The block can be used only in the working screen.

To save tedious painting in the magnifying glass, you can flood-fill a whole area at once by holding the ALT key down while pressing the left mouse button anywhere in the magnified region. The area will be filled with the current drawing color. (Only the enlarged part, not the whole screen.)

DISK OPERATIONS

General

Disk access is gained via the little diskette icons (3.5" and azure blue, naturally) on the menu.



Load a picture



Save a picture



Slide Show (Animation from disk with smooth transitions (morphing))



Disk Info | Delete a File | Virus Test | New Folder | Format



Look for picture data and sprites in files



Look for picture data on disk tracks or sectors

These six functions can be called from the menu or via the function keys F1 thru F6. Most diskette operations call up a file selector box.

The Layout of the File Selector Box

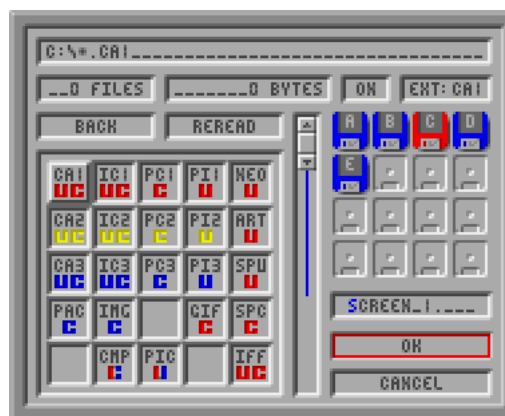
The upper line displays the current path, or the end of it if it is longer than the space provided. The next line contains values for the number of files meeting the search criterion (file extension), the number of bytes in these files, an on-off button for disk verify, and the search criterion itself. The search criterion cannot be edited. Rather, you can choose the current one (e.g. CA1) or *.* during LOAD and SAVE. With LOAD, you can also set the file extension to '???' and the extension of the file you choose determines which picture format Crack Art assumes.

In the next line are two buttons, BACK and REREAD. Use BACK to return to the next higher directory. REREAD causes the directory to be read again.

On the right side are up to 16 diskette symbols labeled A thru P, representing the floppy drives and hard disk partitions on your system. The active drive is red, and all others are blue. Switch between drives by left clicking on the one you want. Clicking the right mouse button on a drive symbol causes a display of the current path, the number of bytes used on the disk, and the number of bytes free on the disk. If you have but one floppy drive and no hard drive, only Drive A is shown, and the imaginary Drive B is not depicted.

Shown below BACK and REREAD are up to 12 file and/or folder names. Behind the files is their length in bytes while folders are marked by FOLDER in blue. With the slider on the right, you can move the window to display other files and folders (in real time!). Clicking on the arrows at the top and bottom of the slider bar causes the same action, but stepwise. A vertical line under the slider shows your relative position in the current folder or disk.

Changing the file format: The PIC-TYPE menu



During LOAD and SAVE operations, right clicking in the file display window allows you the choice of 25 different picture formats. A menu of picture types appears, showing the usual extenders. Each button on the menu is marked with a three-character extender and below the button is a U or a C for uncompressed or compressed. In addition, the three standard resolutions are color coded – red for low, yellow for medium, blue for high, and half red/half blue for all.

Picture Formats

Once you choose a picture format in the PIC-TYPE menu, the customary extender for that type is used for the file search extender. That way, you can use any extender you want and let

Crack Art keep track of the types. For example, an uncompressed picture in Degas format does not have to have the extender 'PI1' to be loaded as a Degas picture. Of course, you must set the search extender to *.* to bring the file name into the selector box. Just click on it and remember that the search extender and the actual extender are two separate critters.

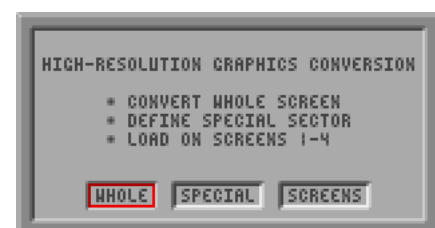
<u>EXT ORIGIN</u>	<u>RESOLUTION</u>	<u>COMPRESSION</u>
CA1 CRACK ART	LowRes	compressed/uncompressed
CA2	MidRes	
CA3	HighRes	
IC1 IMAGIC	LowRes	compressed/uncompressed
IC2	MidRes	
IC3	HighRes	
PC1 DEGAS Elite	LowRes	compressed
PC2	MidRes	
PC3	HighRes	
PI1 DEGAS	LowRes	uncompressed
PI2	MidRes	
PI3	HighRes	
PAC STAD	HighRes	compressed
PIC SCREEN	Low/Mid/HighRes/VGA	uncompressed
CMP F.Mathy	Low/Mid/HighRes	compressed
SPU SPECTRUM 512	320*199	uncompressed
SPC	320*199	compressed
ART ARTDIRECTOR	LowRes	uncompressed
NEO NEOCHROME	LowRes	uncompressed
IFF AMIGA	320*200 4/5/(6)Planes	compressed/uncompressed
IMG GEM	monochrome	compressed
GIF Compuserve	320*200 4 planes	compressed

LOAD - (F1)

Once you find the file you want in the format you want, it is loaded into the current screen if the screen happens to be unoccupied. Otherwise, Crack Art looks for an empty screen and if it finds one, asks you via an alert box whether you want to load into the current screen or the empty screen. If all screens are occupied, Crack Art asks whether you really want to overwrite the current screen. If difficulties crop up during loading, Crack Art tries to load as much information as possible.

SPECIAL HANDLING OF OTHER PICTURE FORMATS AND/OR RESOLUTIONS

LowRes pictures are loaded without any processing or conversion. MidRes pictures are mapped into ten colors and appear exactly as in medium resolution. Monochrome pictures can be converted by any of three ways. After loading, the program asks whether you want to keep the entire picture (WHOLE), only a part (SPECIAL), or divide it among the first four screens (SCREENS). If you choose WHOLE, the picture is rendered in five shades of gray. If you choose SPECIAL, the entire picture is displayed, and you can cut a 320*200-pixel section out of the 640*400 original. The section then becomes a pixel for pixel color version of the selected portion of the original. No gray scale is generated. SCREENS works almost identically, but puts one fourth of the picture on each of the first four screens. In TT Show mode, the complete monochrome picture can be shown.



SPECTRUM 512

SPECTRUM 512 pictures offer still more choices. After the picture is loaded, the user is asked how he wants the color palette matching to proceed.

WHOLE uses the entire picture as a basis for conversion (most used colors are taken).



SPECIAL lets you match rectangular sections of the picture one at a time to get a better match. In the latter case, the picture is shown as a SPECTRUM 512 original in its original colors!!! (Because we program so near to the hardware, this works only on 8 MHz STs!) Regrettably, we had to disable the mouse here in order not to disturb the interrupt-driven construction of the picture. Therefore, you move the crosshairs with the cursor keys. One press moves the crosshairs eight pixels. Shifted cursor keys move the crosshairs one pixel. Mark one corner with the HELP key, then mark the opposite. Everything outside the rectangle is erased and the color palette is adjusted according to what is inside the rectangle. That done, the entire picture is redrawn with this palette. (Every conversion takes three seconds on the ST, while the Spectrum 512 conversion program SPECDEG.TTP needs 16 seconds!!!) In this way, it is possible to hold nearly the original colors in various parts of the picture.

CURRENT uses the current color palette for the conversion. It works best if you build a color palette by hand first, so you have the colors and gradients you want. Regardless of conversion, you will need every byte of memory. Thus, you might lose a block mask definition as Crack Art tries to make more space. The block mode after loading a SPECTRUM 512 picture is always XRAY.

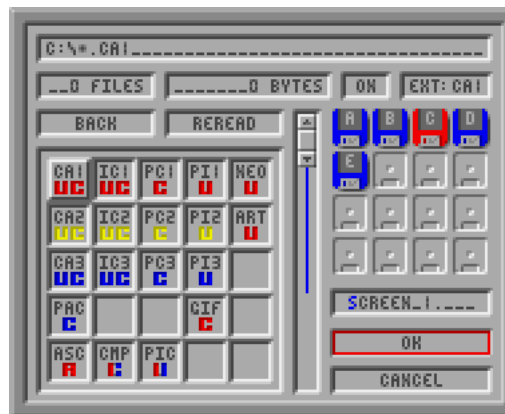


IFF FORMAT

CRACK ART loads IFF packed pictures with resolutions of 320*200 pixels in 16 (4 bitplanes) and 32 (5 bitplanes) colors without problems. Pictures with 64 colors (6 bitplanes) in Amiga Half-Bright mode are not included. The program can convert them, however. (But not quite right. - Roy)

Hold-and-Modify pictures are read with no problems, but the conversion is carried out for the half bright mode and you get junk on the screen. Sorry!

Note: Deluxe Paint for the Atari saves IFF pictures compressed. Normally, this is no problem, but the documentation of this pack format (not standard IFF) is not obtainable. Result: Garbage. Sorry again. To the genius programmers of DP: Great idea kids! Genuinely user friendly, compatible, and with the best outlook for broader acceptance of the format!!!

SAVE - (F2)

If you are ready to save the picture on your screen, pick **SAVE** from your menu and then choose the format from your **PIC TYPE** menu. Choose the floppy drive or hard drive partition as described above. You then have the opportunity, by double clicking on a name in the file selector box, to save under that name, or you can type in a new name. (As already mentioned, extenders are arbitrary, but they should nonetheless at least suggest the format.) If you elect to overwrite an existing file, CRACK ART will ask whether you are sure.

**CONVERSIONS**

Medium resolution pictures are stored in four levels of gray. If you want to save pictures as monochrome, you have the possibilities:



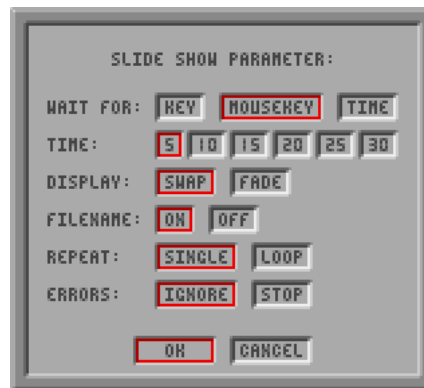
MASKED - One to one storage. Every colored pixel (means: not background) will be stored in a monochrome picture as one pixel.

FOLLEY, **LINEAR**, and **QUADRATIC** are pattern conversion algorithms. The gray scales can be saved as actual patterns (**PATTERN**) or with error diffusion (**RANDOM**). **RANDOM** produces softer transitions.

SCREENS - Like **MASKED** but combines the first four screens into a monochrome picture.

SHOW - (F3)

Slide show down a path you select in the file selector box. Lower folders are recursively (Yes, we can do that, too.) searched. Variable control modes and effects. The selected values will be saved when you **SAVE SETTINGS**.



DISK MENU - (F4)

- | | |
|-------------------------|---|
| 1. DISK INFO | Brief info about the floppy drive(s). |
| 2. VIRUS TEST | Boot virus test of Drive A (and B). |
| 3. DELETE FILE | Here files can be nibbled into the eternal bit heaven. No entire directories, please. |
| 4. CREATE FOLDER | Self explanatory |
| 5. FORMAT DISK | Should you ever find yourself in the unfortunate situation of having no disk for your latest masterpiece, help is here! |



CRACK ART can format either single- or double-sided disks with 80 or 82 tracks. First, the type of boot sector is determined:

Anti-Virus: When you use the computer with this disk in the drive, a small message on the screen indicates NO boot sector virus. There is also a small background sound.

Bootdump: BOOTDUMP runs in all ST resolutions. (A gem of 68K code...) With the help of this boot sector, one can grab up to 800 KB out of running (not reset proof) programs. (For Christians: We grab, we do not steal! Therefore, no lawbreaking. Besides, I'm an atheist. JayBee) Operation is simple:

- Load the program (Ball game with Super Graphics, but not Xenon 2. It can even cope with the NMI (Non-Maskable Interrupt)).
- Play till the desired graphic appears.
- Insert the BOOTDUMP disk.
- Press RESET and hope that the program is not reset-proof or doesn't clear memory.

Now you have the 'What's happening, machine?' before you: The computer reboots and on the screen appears graphically depicted data garbage from the start of memory. With the cursor keys you can scroll up and down through memory. The 'run width,' or amount of memory to save to disk, is specified with the number keys. The current memory address, from which point the save begins, is displayed on the top edge of the screen, along with memory start and end (everything in HEX). Press RETURN to begin the dump to (we hope empty) disk. If the disk is already half full, nothing will be destroyed. Rather, a file will be written to the remaining space. Hit ESCAPE to abort.

With the CRACK FILE function, one can then take the data apart in CRACK ART. Iff you have a hard disk, you should copy the file (It can be up to 800 KB) to it first, to save load time. Mid-

and Hi-resolution screen data can be converted via the MID and HI options in the EFFECTS menu.

Make your choices in the dialog box. Once you have made your choices, formatting begins with no more questions. A graphic (prepare to be surprised) depiction of the process appears. If disk problems occur, their form is reported, and control returned to the user. The disk name is generated from the format mode:

```
CRACKART.SS   Single sided, 80 tracks
               .SS+  Single sided, 82 tracks
               .DS   Double sided, 80 tracks
               .DS+  Double sided, 82 tracks
```

An antivirus program is written in the boot sector. On bootup, it prints the message, 'ANTI VIRUS BOOTER and emits an unmistakable acoustic signal. (One does not have to stare stupidly at the screen anymore while he daydreams.)

Comments: Keep a formatted bootdump disk handy. If the computer crashes, pop the disk in and hit RESET... and save. Graphics are that simple to reconstruct.

CRACK FILE - (F5)

The graphic file monitor. It can search through files looking for graphic data and palettes.

LOAD - (RETURN KEY)

Load graphics or color palettes relative to an offset.

Data will be loaded starting at the position denoted by OFFSET and displayed on the screen. When PIC is active, up to 32,000 bytes will be loaded to the screen. With PAL active, Crack Art loads 32 bytes into the color palette and shows that. Loading goes approximately twice as fast from even offsets as from odd.

SAVE

Save data to disk.

Screen data, starting at the data position denoted by OFFSET, will be saved to disk.

Remarks: For safety's sake, SAVE requires a double click.

VIEW

Display the entire screen

When you click on the VIEW button, the menu disappears, and the entire screen is visible.

EXIT - (ESC KEY)

End Crack File functions

With this, you leave the Crack File functions. The name of the file you were working on and the parameters you specified are saved. The next time you crack a file, the name appears as the default.

SEARCH (SPACE BAR)

Advance OFFSET and load data

When PIC mode is active, the OFFSET pointer is advanced to the data just after that displayed on the screen, and that data is loaded into screen memory. In PAL mode, Crack Art searches screen memory for valid color palettes. A color palette is valid when 16 successive nonzero words occur and each of them fits in a \$777 mask [That's hexadecimal, not high-priced Halloween. Exactly what 'fits' means, I do not know. An experienced programmer should. (Means: values of RGB are coded in three nibbles (one hex digit = 4 Bits) with no entry bigger than 7 ==> no STE palette. -Detlef)] When a valid color palette is found, it is immediately activated. If you press SEARCH again, the next palette is sought and activated if found. Return to the original palette with UNDO.

WIDTH

Specify the line length

This value specifies how many bytes to read per line. Default is 160 bytes, the width of the screen. Eight bytes per line would correspond to a 16-color sprite, 16 bytes wide.

OFFSET

Set the file pointer

Here you can set the file pointer to the start of the next data to be loaded. In addition to directly specifying OFFSET, you can press the right mouse button. The menu disappears and a crosshair appears. With this and the left mouse button, you can pick the start point (i.e. the new OFFSET) off the screen. Press the right mouse button again to exit.

CRACK DISK - (F6)

The graphic disk monitor. It can search through disks looking for graphic data and palettes. Before using it, check the parameters (sides, tracks, sectors) under DISK INFO in the DISK MENU. Reading invalid sectors or tracks causes errors which, though ignored, cost time. The method used by CRACK DISK is complicated and will not be explained here. The function is, however, fully implemented. (The function works in the same way as CRACK DISK, but uses sides, tracks, and sector instead of file offset. It could only be necessary to use it, when your disk has crashed, and you like to rescue some (unpacked!) data.)

EFFECTS (FX)

The EFFECTS button in the menu offers access to additional picture processing functions, which cannot be reached from the keyboard. (42 + 6. The answer to every question! (+6))

After selection of the button, the current screen is displayed. On its right side is the menu for the various effects. The menu is divided into names and numbers. The numbers represent the

working screens. Clicking on a number switches you to that screen. Titled fields initiate functions. Holding the SHOW field with the left mouse key displays the entire screen.

For all functions, the UNDO key undoes the last operation. The working screen and color palette are restored. However, if you switch to another screen before trying to UNDO, you're committed. All changes are permanent once you switch.

The Effects menu is redisplayed on completion of a function.

Return to the main menu with the right mouse button.

Note: In the following text, the terms 'surrounding pixels,' 'neighbors,' etc. are used. The indicated pixels lie in a 3x3 matrix around the pixel to be processed, which is marked with an X, and are selected analogous to one of the two matrices below. When, for example, five pixels are pulled in for processing, a glance at the diagrams below will help you understand the concepts of the center point and neighbors and the operations Erosion, Median, and Dilatation.

ANTI4	ANTI8	ROUND						
REL 0	REL N	MINI						
RES 2	RES 4	PIXEL						
OR	NASH	SWEEP						
OUT 4	OUT 8	EDGE						
MED -	MED	MED +						
ERO	CONTR	DILA						
UNZAG	HSAT	USAT						
LIGHT	CON 4	CON 8						
DARK	BUT 4	BUT 8						
RED	GREEN	BLUE						
CYAN	MAGNT	YELL						
GREY	FADE	SEPAR						
MID	HI	TILE						
HFREE	H SIM	H ZIG						
UFREE	U SIM	U ZIG						
U	-001	-002	-00					
S	±000	±000	±000					
R	+001	+002	+00					
1	2	3	4	5	6	7	8	9

```
0 1 0
2 X 3
0 4 0
```

4 Neighbors

```
1 2 3
4 X 5
6 7 8
```

8 Neighbors

Effects Functions

ANTI 4

Function: Antialiasing function using four surrounding points.

Description: After ANTI 4 is selected, the entire screen is processed from top to bottom. Multiple applications of the function bring other results.

Remarks: Since the function considers only four surrounding points, it works less finely than the ANTI 8 function. Heavy application of ANTI 4 to a detail-rich picture is a kind of tone separation. The antialiasing function works best on color palettes with the maximum possible levels of a color.

ANTI 8

Function: Antialiasing function using eight surrounding points.

Description: Identical to ANTI 4.

Remarks: The consideration of eight surrounding points yields a somewhat finer effect. The algorithm works somewhat more slowly toward the above-mentioned tone separation. On thinly populated pictures, though, you will see a rapid development of the details toward the surrounding color.

ROUND

Function: Rounds corners off.

Description: The ROUND function resembles the ANTI 4 function except that it operates only on pixels that form corners in the usual sense. Repeated application makes sense.

Remarks: The function is handy for enlargements, where you want to smooth out the lumpiness. Repeated application tends toward 45-degree corners.

REL M

Function: Transforms a color picture into a relief picture with three levels of gray.

Description: REL M process the picture diagonally from the lower left, using a very simple process: The higher the color number of the pixel, the longer the shadow. To be sure, we allow for such trivialities as plateau formation and sudden edges. The result has a palette of eight shades of gray, the minimum to produce the effect properly. Shadows are shown in color 0 (black), plateaus in color 4 (gray), and highlights in color 7 (white).

Remarks: The results obtained with this function depend in strong measure on the arrangement of the colors in the original palette. The best results come from an arrangement according to brightness. Whether the relief bulges out or dishes in can be controlled. See SORT, INVERT, and EXCHANGE in the color menu. Since the color palette contains intermediate levels in addition to the three required colors, it can make sense to run the result through one of the antialiasing functions. That usually produces a better-looking picture. You can control the shadow direction by flipping the screen first. (That can be done with the cursor keys via the Effects menu.)

REL C

Function: Produces a pseudo relief.

Description: The function works similarly to REL M. Pixels are darkened where shadows fall, highlights are brightened, and plateaus stay the same.

Remarks: Operation is identical to REL M.

MINI

Function: Create a mini picture.

Description: The picture is shrunk to an 80x50-pixel gray scale picture.

Remarks: This function is used in the main menu to produce the overviews.

RES 2

Function: Reduces the resolution of a picture to one fourth.

Description: Out of 2*2-pixel regions, the function takes the color of the upper left pixel and replaces the colors of the other three with it. Repeated applications bring no further change.

Remarks: This function yields amazing effects in combination with REL M, REL C, OUT 4, OUT 8, and EDGE (Which see).

RES 4

Function: Reduces the resolution of a picture to one sixteenth.

Description: Results as from RES 2, only more abstract.

PIXEL

Function: Coarsens the resolution to a 2*2-pixel basis.

Description: Results are similar to those from RES 2, except that less information is lost because nine original pixels go into the calculation of the new 2*2-pixel rectangle.

Remarks: Only unimportant changes result from repeated applications. Quartering this picture with SIZE is no better than HELP with SIZE. (It's logical, no?)

OR

Function: Produces a mask from the graphic data.

Description: All pixels with a non-zero color number are set to color 15. Repeated application yields no further change.

Remarks: This function is handy for masking out particular graphic regions with the help of the block operations.

MASK

Function: Produces an inverse mask from the graphic data.

Description: All pixels with a color number of zero are set to color 15 and all other pixels are erased. The result is the inverse of OR. Repeated application toggles the mask back and forth. Two applications are identical to one OR.

Remarks: Use is just like OR.

SEPARATE

Function: Erases pixels which match those in a second screen.

Description: If SEPARATE is clicked, the symbols for all occupied screens are shown reversed in order to show that further choices follow. Clicking on one of the screen symbols selects that one to compare against the active screen. The operation is carried out according to the rule, "Erase all pixels from the active screen which agree with the corresponding pixel on the comparison screen." The comparison screen is not altered. The pixel count on the active screen will not increase.

Remarks: The function is handy during development of graphics to see what was changed from one step to the next. Partial antialiasing etc. are easy afterwards.

OUT 4

Function: Leaves the boundaries of filled areas, referring to four neighbors.

Description: The inner regions of filled areas are erased, leaving only the boundaries.

Remarks: Complementary function to EDGE. Also produces astounding results in combination with RES 2, RES 4, PIXEL, REL M, and REL C.

OUT 8

Function: Leaves the boundaries of filled areas, referring to eight neighbors.

Description: Same as OUT 4.

Remarks: Produces astounding results in combination with RES 2, RES 4, PIXEL, REL M, and REL C.

EDGE

Function: Erases the boundaries of filled areas.

Description: The edges of filled areas are erased. Thus, it is possible to separate adjacent areas.

Remarks: Complementary function to OUT 4.

MED-

Function: Median filtering with slight darkening.

Description: The RGB values of five pixels (see diagrams before ANTI 4 and ANTI 8) are sorted separately and the second darkest are used.

Remarks: Median filters are used in digital image processing to reduce disturbances. Belongs to the group ERO, MED-, MED, MED+, DILAT.

MED

Function: Median filtering.

Description: The RGB values of five pixels are sorted separately and the middle values are used.

Remarks: Belongs to the group ERO, MED-, MED, MED+, DILAT.

MED+

Function: Median filtering with slight lightening.

Description: The RGB values of five pixels are sorted separately and the second brightest are used.

Remarks: Belongs to the group ERO, MED-, MED, MED+, DILAT.

ERO

Function: Erosion.

Description: The RGB values of five pixels are sorted separately and the darkest values are used.

Remarks: Belongs to the group ERO, MED-, MED, MED+, DILAT.

CONTR

Function: Contrast enhancement.

Description: If a pixel is lighter than its eight neighbors (see diagrams before ANTI 4 and ANTI 8), it is darkened and vice versa.

Remarks: Makes barely perceptible color differences easier to see.

DILAT

Function: Dilatation

Description: The RGB values of five pixels are sorted separately and the brightest values are used.

Remarks: Belongs to the group ERO, MED-, MED, MED+, DILAT.

UNZAG

Function: Rounding edges off.

Description: The UNZAG function resembles the ROUND function. The operation here, however, is restricted to the pixels which form a 45-degree edge.

Remarks: Very limited applicability.

H-SHAT

Function: Horizontal fraying of the picture.

Description: The positions of each pair of horizontally adjacent pixels are switched.

Remarks: This function is reversible. See V-SHAT.

V-SHAT

Function: Vertical fraying of the picture.

Description: The positions of each pair of vertically adjacent pixels are switched.

Remarks: This function is reversible. See H-SHAT.

USER

Function: Apply a user-defined filter to the picture.

Description: Values between -100 and +100 can be loaded into a 3x3 matrix with the mouse. This matrix can then multiply the RGB values of a pixel. Load the values by selecting a number, holding the left mouse button down, and moving the mouse horizontally. Double clicking on a value sets it to zero.

Remarks: Slowest, but most flexible of the Effects functions. Try, for example:

Enhance Detail: Sobel:

0	-1	0	-1	-2	-1	-1	0	1	
-1	10	-1	0	0	0	and	-2	0	2
0	-1	0	1	2	1		-1	0	1

FADE

Function: Blending of two pictures.

Description: Options as in Separate. Up to 50 percent of the second picture will be blended into the first.

Remarks: With multiple applications, the second picture becomes clearer and clearer. Because of the integer arithmetic used in the calculations, however, the process may stagnate.

SWEEP

Function: Erase isolated pixels.

Description: Pixels whose eight neighbors have a common color, distinct from the color of the first pixel, are given the color of the eight neighbors.

Remarks: Use during image processing to filter out single odd spots.

LIGHT

Function: Brighten the picture without changing the color palette.

Description: The brightest color in the color palette is selected and the pixels in the picture are brightened in the direction of this tone.

Remarks: Opposite of DARK.

DARK

Function: Darken the picture without altering the color palette.

Description: The darkest color in the color palette is selected and the pixels of the picture are darkened in the direction of this tone.

Remarks: Opposite of LIGHT.

GREY

Function: Converts a picture into eight levels of gray.

Description: The colors are converted into gray values according to their brightness and the pixels set accordingly.

Remarks: You can also mix pictures with different color palettes.

RED

Function: Pull the red out of a picture.

Description: The red values of the pixels are presented as a grayscale picture. The color palette ends up as gray and red.

Remarks: Belongs to the group RED, GREEN, BLUE, CYAN, MAGENTA, CYAN, YELLOW.

GREEN

Function: Pull the green out of a picture.

Description: The green values of the pixels are presented as a grayscale picture. The color palette ends up as gray and green.

Remarks: Belongs to the group RED, GREEN, BLUE, CYAN, MAGENTA, CYAN, YELLOW.

BLUE

Function: Pull the blue out of a picture.

Description: The blue values of the pixels are presented as a grayscale picture. The color palette ends up as gray and blue.

Remarks: Belongs to the group RED, GREEN, BLUE, CYAN, MAGENTA, CYAN, YELLOW.

CYAN

Function: Pull the cyan out of a picture.

Description: The cyan values of the pixels are presented as a grayscale picture. The color palette ends up as gray and cyan.

Remarks: Belongs to the group RED, GREEN, BLUE, CYAN, MAGENTA, CYAN, YELLOW.

MAGENTA

Function: Pull the magenta out of a picture.

Description: The magenta values of the pixels are presented as a grayscale picture. The color palette ends up as gray and magenta.

Remarks: Belongs to the group RED, GREEN, BLUE, CYAN, MAGENTA, CYAN, YELLOW.

YELLOW

Function: Pull the yellow out of a picture.

Description: The yellow values of the pixels are presented as a grayscale picture. The color palette ends up as gray and yellow.

Remarks: Belongs to the group RED, GREEN, BLUE, CYAN, MAGENTA, CYAN, YELLOW.

MID

Function: Convert from medium to low resolution.

Remarks: See CRACK FILE and CRACK DISK.

HI

Function: Convert from high to low resolution.

Remarks: See CRACK FILE and CRACK DISK.

TILE

Function: Tessellate the screen with a rectangular section of the picture.

Description: After you click on the menu field, use the crosshair which appears to cut a rectangle out of the picture (exactly like cutting out a block). Position and size of the block appear as usual in the lower left corner. The cutout is then used like tiles to cover the entire screen. Cutouts along the right edge and bottom of the screen are clipped if necessary.

Remarks: This routine is very well suited for developing animations from a basic pattern. It is also applicable to the drawing routines which use blocks as fill patterns.

H-FREE

Function: Horizontal displacement of screen lines.

Description: The button enables a free displacement of individual screen lines through use of the cursor. Once the button is clicked on, a vertical reference line appears in the middle of the screen. This is your zero. Click and hold on any pixel in this line and you can drag it left or right as you wish. Wherever you leave this pixel is the new zero for that line. Work your way up and down the screen, setting offsets where you will. When you click on the right mouse button, the displacements are carried out and you find yourself back in the effects-menu.

H-SIN

Function: Horizontal sinusoidal displacement of screen lines.

Description: Once this button is clicked on, a vertical reference line appears in the center of the screen, along with a frame whose dimensions are displayed in the lower left corner. Use the frame to set the size of a half-period sine wave (Think of a fat man's belly just touching the three sides of the frame.). Click the left mouse button to execute. The result is a bowed or wavy picture.

H-ZIG

Function: Horizontal triangular (zig-zag) displacement of screen lines.

Description: Analogous to H-SIN except that here a sawtooth wave is used. The result is a kinked or zig-zag picture.

Remarks: Choosing a tiny frame, say 1*1 pixel, causes light smear effects which look different from those created with H-SHAT.

V-FREE

Function: Vertical displacement of screen columns.

Description: Analogous to H-FREE.

V-SIN

Function: Vertical sinusoidal displacement of screen columns.

Description: Analogous to H-SIN.

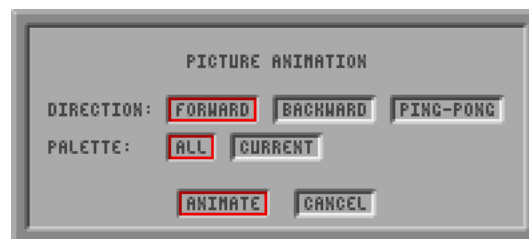
V-ZAG

Function: Vertical triangular displacement of screen columns.

Description: Analogous to H-ZIG.

SCREENS

Here we finally explain the blue light diodes. (Do you have any idea how expensive blue LEDs are? Up to DM20 [approx. \$13] each! But only the best is good enough for CRACK ART!) When CRACK ART loads, all available screens are marked with a blue LED. Double-clicking on a blue LED deactivates the corresponding screen, removing it from the animation list. Double-clicking again turns the screen back on and puts it back in the list. After you select the animation icon in the menu, an alert box appears, through which you can control the direction of the animation.



FORWARD: The screens follow in ascending order.

BACKWARD: The screens follow in descending order.

PING-PONG: The animation runs back and forth, over and over.

PALETTE: ALL or CURRENT

The speed of the animation can be controlled via the horizontal position of the mouse cursor. The fastest sequence occurs with the cursor against the left edge of the screen. If you select ALL palettes, the computer switches palettes to match each picture. If you want to use only one palette, select CURRENT.

The TT throws a monkey wrench into the works here. If you drive the animation too fast, the picture sways. Therefore, foot (hand) off the gas.

SPRITES

LOAD: Load the definition file of a sprite sequence.

SAVE: Save a definition file.

INFO: Display the sequences and screens used.

SIZE: Alter the size of the sprite buffer.

The control and definition of the sprite buffer are covered in the color menu. Sprites are the coordinates of rectangles on the working screen. The graphic data depend only on the picture and are not altered. That has the disadvantage that you must load the picture (possibly several), which contains the graphic, yourself. An advantage, though, is that the graphic can be swapped around. Should you produce many animations of the same size but with different graphics, you need only to exchange the picture. INFO tells you which pictures are used in an animation.

You can save the sprite data for the current sequence as assembler data with ASC in the PIC-TYPE menu of the File Select Box.

RELAX

A Relax game with the extension .GAM is loaded from the GAME folder at every start of Crack Art and the button appearance will be redefined. If no game is found, a neutral relax button appears.

We have let our infantile drives have free run here. The graphics were naturally created with CRACK ART.

All games can be run separately via the program GAMERUN.PRG.

ASTROID

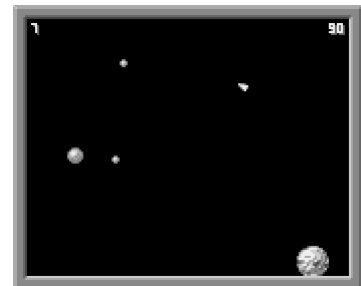


by 'Roy' Röttger

TECHNICAL DATA:

6930-byte program

Maximum 32 stones, 16 guns, and one spaceship on the screen. Spaceship motion is computed in a fixed-point routine. The collision routine interrogates a diamond shape rather than a rectangle. This makes the sprite form more realistic. The speed is throttled to a playable level. The ST serves as the standard. When there is a lot of activity on the TT, the game runs faster instead of slower! Sorry about that.



CONTROL:

Joystick (implemented by JayBee):

Left, right, forward, fire.

Keyboard (in the same order as above)

Left Shift, Alternate, Control, Right Shift

ESC to end the game.

REMARKS:

No extra lives.

Game over when spaceships are all gone.

GAMEPLAY:

Shoot everything in sight except the scoreboard.

BRAIN (Master Mind)

by 'JayBee' Borchers

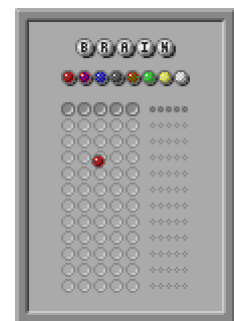
TECHNICAL DATA:

6676-byte program

Mouse control, moving text, and overblend effects.

CONTROL:

With the left mouse button, set colored stones onto empty spaces or take new colors from occupied spaces. With the right mouse button, you can get suggestions for solution. Doubled colors are a no-no.

REMARKS:

Game over when solved, not solved, ESC key and then mouse.

GAMEPLAY:

Try to find the solution within the allowable moves. Doubled stones are not allowed. In the example, available colors are marked with white stones and properly placed colors with black stones.

T-TRIS

by 'JayBee' Borchers

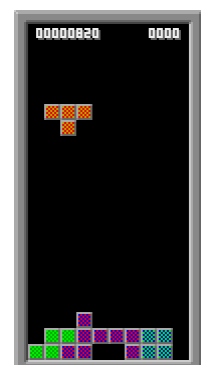
TECHNICAL DATA:

5772-byte program

Keyboard control, some sound (Graphics interest me more.)

CONTROL:

Left - Left arrow, 4 (ten spaces)
 Right - Right arrow, 6 (ten spaces)
 Turn - Up arrow, 8 or 5 (ten spaces)
 Down - Down arrow, SPACE, RETURN, ENTER, 2 (ten spaces)
 Quit - ESC



REMARKS:

Game over when you pile the stones up to the upper border of the playing area or ESC.

GAMEPLAY:

Try to build as many filled rows as possible. These are then automatically cleared.

PACNAM

by 'JayBee' Borchers

TECHNICAL DATA:

14786-byte program

Keyboard control

CONTROL:

Left - Left arrow, 4 (ten spaces)
 Right - Right arrow, 6 (ten spaces)
 Up - Up arrow, 8 (ten spaces)
 Down - Down arrow, 2 (ten spaces)
 Pause - SPACE
 Quit - ESC

REMARKS:

Game over when you lose all five lives or press ESC.

GAMEPLAY:

Try to eat all the yellow dots on the playing area and do not let the ghosts wipe you out. Eating a big dot causes the ghosts to flicker. During this time, they can be eaten.

MINE

by 'Roy' Röttger

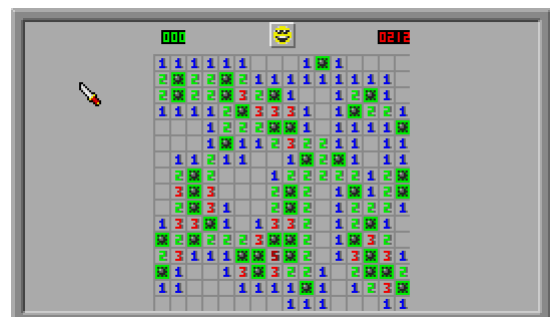
TECHNICAL DATA:

6298-byte program

Mouse control

CONTROL:

Use the left mouse button to reveal fields and the right mouse button to mark fields as mined. If you right click on an already revealed field, any unmarked neighbors will also be revealed. Should one of these unmarked neighbors be mined - BOOM!



REMARKS:

Game over with ESC

GAMEPLAY:

Try to identify all hidden fields: Empty fields must be revealed; mined fields must be marked.

PRINTER



Loads a printer driver or calls an already loaded driver. Crack Art includes a number of drivers in the PRINTER folder. Marked with the suffix .CAP (for Crack Art Printer), drivers are loaded via the PRINT button. Each routine contains entries for name, source- and target size of the picture, and its own icon which replaces previous printer symbol. The header construction is explained in the assembly listing of the driver. (Is there one on disk?)

The size of the driver buffer is set statically when the program starts. Should a driver not fit in the buffer, set the size to suit. (See next paragraph to learn how.) Altering the size of all buffers has the advantage that you might save the space that gains room for one more working screen (only 1 MB computer).

If a printer driver is already loaded, there appears (for the drivers we have written so far) a new icon, substituting for the printer symbol. Clicking on the icon brings up a dialog box which gives the driver name, output quality, and method of operation. From the options in this box, you can print the current picture, load another driver, adjust the size of the driver buffer (Save Settings and restart for it to be effective), or abort.

Should it happen that all you get is PRINTER NOT READY when everything should work, read the README.TXT in the PRINTER folder in which CLEANER.CAP is located. With this driver, you can print with the printer offline and set it on line in the process (or after an error. Try it.). With the HP Deskjet 500 (color), this problem appears regularly due to hardware problems (and lazy programmers.)

SCAN



A scanner driver can be loaded and used here. (We added this feature, but we do not know if a scanner module has been written for it. There was a driver for a color scanner at the Dusseldorf Atari Fair in 1992, but we did not write it. Our scanner module calls a driver which is already loaded, e.g. at boot up. If you do not have a scanner that works with Crack Art, you don't have a scanner driver either. Therefore, nothing (almost) happens when you call the scanner module. (The connection passes a pointer to the structure described under ACC. You can therefore (mis)use it for your own purposes.) If a scanner driver is loaded, you can call it directly through the main menu with SPACE without using a dialog.

ACC



Call accessories. Only if you have more than 1MB RAM should you load any (large) accessories.

CRACK ART allows communication with accessories via the SCRAP buffer. This buffer area is normally used to pass file names, therefore for compatibility reasons, this communication passage can be turned on and off. That is, whatever is in that area when Crack Art starts will be written back. The (de)activation of the SCRAP connection is accomplished via the menu item FILE / SCRAP INTERFACE.

A ten-byte string, containing 'CA' in the first two bytes, is written to the SCRAP buffer. The remaining eight bytes represent a pointer to a 256-byte structure, which is coded in ASCII-HEX. At this address, the following structure is found:

- WORD NumberOfScreens	; Number of work screens (1-9)
- WORD CurrentScreen	; Current work screen (1-9)
- LONG CurrentScreenAdr	; Start address of the current work screen
- LONG CurrentPaletteAdr	; Start address of the current working palette
- LONG WorkScreenAdr_1	; Address of the first screen available for ; Crack Art's own use. This block of memory ; will be used again by Crack Art after the ; accessory finishes.
- LONG WorkScreenAdr_2	; Address of the second screen (see above)
- LONG WorkScreenAdr_3	; Address of the third screen
- LONG BlockScreenAdr	; Address of the block screen, on which the ; current block is found
- LONG BlockPaletteAdr	; Address of the block palette
- LONG BlockMaskScreenAdr	; Address of the block mask screen, on which ; the current block mask is found (fully ; colored, i.e. all planes are occupied)
- WORD BlockW	; Width of the block in pixels
- WORD BlockH	; Height of the block in pixels
- LONG ScreenAdr_1	; Start address of the first work screen
- LONG ScreenPaletteAdr_1	; Start address of the first palette
- LONG ScreenAdr_2	; Start address of the second work screen
- LONG ScreenPaletteAdr_2	; Start address of the second palette
.	
.	
.	
- LONG ScreenAdr_9	; Start address of the ninth work screen
- LONG ScreenPaletteAdr_9	; Start address of the ninth palette

Addresses of unused screens and their palettes are set to zero.

Additional transfer parameters as of Version 1.20 TT+

- WORD CurSrcW	; Width of the current screen (currently fixed ; at 320)
- WORD CurSrcH	; Height of the current screen (currently ; fixed at 200)
- WORD CurPalColors	; Colors in the current screen (currently ; fixed at 16)
- WORD CurPalBits	; Color depth in bits per RGB of the current ; screen (currently fixed at 3 -> 512 colors)
- WORD BlockMode	; Block display mode (is read) (0 = X-RAY, ; 1 = OBJECT, 2 = BLOCKED)

Additional transfer parameters as of Version 1.21 TT+

- LONG Processing	; Address of a slider display routine
- BYTE 124-255	; reserved, always 0

Through these addresses, the possibility exists to use your own graphics routines on the screen.

Other programs can be run from the FILE menu by means of EXECUTE *.PRG OR EXECUTE *.TOS. See the sample listing in the folder SCRAPACC.

KEYBOARD MAPPING

The keyboard is loaded with the essential functions to permit work without having constantly to refer to the menu screen.

Most functions are assigned to keys which match their initials. The functions which carry out fill functions are assigned to keys from Z downward. In the menus, the equivalent keys are marked in red.

Key Assignments in the Menu

D Draw	1-9	Screens
P Point	0	Superview
L Line	F1	Load
K K-Line	F2	Save
F Frame	F3	Show
C Circle	F4	Disk Menu
E Eraser	F5	Crack File
A Airbrush	F6	Crack Disk
R Rays	F7	Print
W Polygon	F8	Scan
X Box	B	Brush/Block toggle
Y Disc	(X-Ray
T Text)	Blocked
O Outline	/	Free Move
Z Fill	*	Edge Stop
S Smear	ESC	Switch to working screen
M Mirror	HELP	Just for Joke
I Spline	SPACE	Scanner Hotkey

Keyboard Assignments in the Work Screen

The keys for the drawing functions are identically assigned here. In addition, you can switch among working screens and specify block parameters.

1 - 9	Switch among the working screens
SPACE	Color menu on/off
UNDO	Undo last operation
F1-F10 (+Shift)	Magnifying glass (loupe) 3*3 to 12*12
Cursor left/right	Choose drawing color.
Cursor up	Flip screen vertically
Cursor down	Flip screen horizontally
Cursor+Block	Flip block in corresponding direction
ESC	Cut out rectangular block.
Control+ESC	Cut out circular or elliptical block.
Shift+ESC	Cut out block with lasso.
Alternate+ESC	Cut out block with fill function.
BACKSPACE	Rotate block 90 degrees clockwise (fast)
RETURN	Set block to current color palette

HELP	Minimize block spread (CUT). Does not work with OBJECT.
Clr/Home	Center block on screen
Control+Clr/Home	Center block horizontally on screen
Insert	(TT only) Reduced display of the current screen in TT middle resolution.
(Block XRAY
)	Block BLOCKED
/	Block FREE MOVE
*	Block EDGE STOP

Drawing Functions:

+* Draw	D	= Freehand drawing
+ Eraser	E	= Eraser
+ Point	P	= Single points
+ Line	L	= Draw single line
+* K-Line	K	= Polyline
+ Rays	R	= Rays
+ Frame	F	= Rectangle
+* Circle	C	= Circle and Ellipse
* Airbrush	A	= Spray can
+* Box	X	= Filled rectangle
+* Disc	Y	= Filled Circles and Ellipses
+* Polygon	W	= Filled polygon
* Spline	I	= Two different spline algorithms and one curve routine
* Smear	S	= Smear pixels on the screen
* Outline	O	= Add borders to lines and areas
+* Fill	Z	= Fill a screen region
* Mirror	M	= Draw with symmetry axes
* Text	T	= Put text on the screen

The functions marked with a plus sign work also with blocks. The block then serves as a paintbrush, or for functions that operate on surfaces, as a fill pattern using the current pattern (See also Color Menu).

The functions marked with an asterisk (*) control parameters which can be changed after double-clicking on the icon. The letter after the function name is the keyboard equivalent of the menu icon.

A colored border around the icon shows that it has been selected. The permissible MIRRORed drawing operations are automatically recognized and corrected if necessary. Drawing functions which require several operations are cancelled or ended with the right mouse button.

HAVE YOU EVER SEEN?

- That the operating system bug which left the top two screen lines empty when drawing filled objects has been overcome?
- That the mouse casts a shadow when it finds itself over a menu shadow?
- That the floppy lamps dim rather than blink at the end of information?
- That the borders of the selected icons are not of uniform color, but rather are darker on the shadow side?

- That the mouse throws a shadow on the borders, too?
- That all icons were antialiased by hand?
- That only as many disk drives as are actually attached are shown? (No virtual B through disk swaps.)
- That even the Block Button has a shadow?
- That the day of the week is correctly calculated from the date? [Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday]
- That bad disk sectors are shown in dark red during formatting? (And it means absolutely nothing because I don't flag them in the FAT. Therefore, kismet and into the trash with it! JayBee)
- That empty screens are marked with a graffiti-looking EMPTY SCREEN?
- That you might finally use up the ribbon on your color printer? (Except for lasers, Desk Jets, and similar great holes into which one can throw his entire savings...)
- That every item in 'HAVE YOU EVER SEEN' begins with 'That?'
- That the letter, which waits in front of you with our address on it, ready to be stuffed with 30 Marks [\$20], still has not been stamped?

THE ULTIMATE END - YOU DID IT!

PS: There is much to do; we are doing it to you!