GROSSER SPIELETEIL

Markt&Technik

Juni 1991

öS 60,-/sfr. 7,- Lit. 7400/hfl 9,-/fmk 22,- DM 7,-

64'er

6|91 DAS MAGAZIN FÜR COMPUTER[illegible]S

Bauanleitung

C64 IM URLAUB

- Weg von der Steckdose

Pocket-Writer 3.0

Das beste C64-Textprogramm

Listing des Monats

Autokosten-Manager

- Geld sparen beim Autofahren

64'er-Projekt

C64-Meßlabor

- universell ausbaubar
- 1. Folge: Wetterstation

SPIELE

TESTS: Creatures • Super Cars • Loops • Turrican II

EVERGREEN: David's Midnight Magic

NEUHEIT bei Turrican II: Parallax-Scrolling

LONGPLAY: Bard's Tale

Listing 2. »Window 1.3« erzeugt umrahmte Fenster mit dem C128

ACHTUNG NEUER MSE (auch auf Diskette, s. S. 51)

```
"window 1.3"                    1300 14b5
-----------------------------------------
1300: ud7h k5ui 7rb6 wjh7 pw5j rhde ez
130f: 6vq7 ih77 vg4i c6dh zc35 m56f ax
131e: 6wem a3ui 7bb6 ujnx pw4z r7de a3
132d: 6rtr iao4 thbj 77eq 6jh6 ybfp ed
133c: 6gso wyw4 ykho ejii tclh k5td gs
134b: 6nrn kp77 571a qioz mdth k5ue c4
135a: 6mtp aao2 qc77 atgn gbr5 oyof bj
1369: qurq hnee 3sr5 kckg bplj k5se 7q
1378: 3zb6 uio2 md7h k54i 7bfz dd4i 7a
1387: 7bfp a63m lxjj k5te 6rr6 wao4 b4
1396: isda gjh7 qura ikcd brtp cdoz em
13a5: ybln xy7t yurq itgr cbr6 tzih cm
13b4: pw4j k52i 7bb6 w2sf brvv lefm es
13c3: 13jm ats7 dblq fnee 6stp aaoz ab
13d2: tw55 sude 6ntp ackf brtz acmb bk
13e1: blpi rdyx tw5n rjle 6jr6 2zh7 bt
13f0: pw4z r7dm lxjj r7lm thir afhs ae
13ff: gbr6 yzii pw4j k6oi zrb6 wsse dh
140e: bsgf nee7 7bt6 adoz ug7l qdoz bm
141d: yqrq itgx ugwi c5ub 77lj k5si ef
142c: ebb6 uio2 md7h k547 7bt3 edoz aj
143b: ufpl qdoz yqrq itgx ugai c5wh gt
144a: 4qsq itfy t77j s2lq 6jt3 ardq gh
1459: 6kff jefp 6bt6 2doz 17nq xdpq gl
1468: yd7o 77zl 771z r7de 6jtr aao2 76
1477: ud7h k6ei 7rb6 2hpd t77k c5tq 7d
1486: 6rdm a5of 6oso 2rvp 5jtp aaoz cs
1495: udrh k54i 7bb6 yjnx pw5z dae7 b7
14a4: 7bx6 udo3 qcho syw2 3253 utgr ei
14b3: 1777 a666 666p 7777 7c66 6666 ei
```

© 64'er

C-64-Raffinessen

Linien ziehen

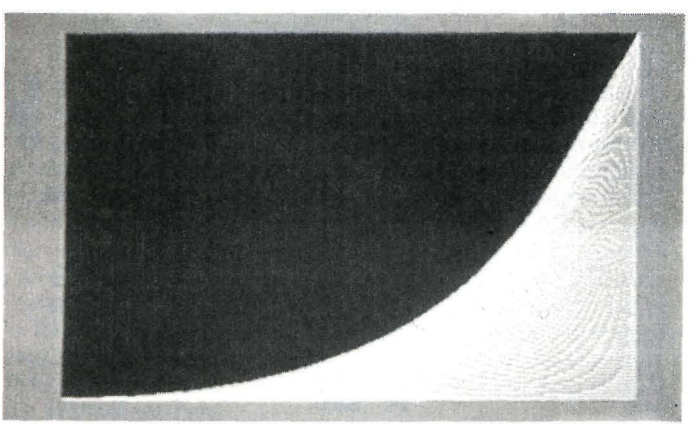
Als Lösung unseres Wettbewerbs präsentieren wir die schnellste Routine zum Zeichnen von Linien auf dem Bildschirm. Der Bresenham-Algorithmus macht's möglich.

Mitmachen – mitgewinnen

Gesucht: Ein Programm, das alle Bildschirmausgaben parallel auf dem Drucker mitprotokolliert, d.h., alles, was auf dem Monitor erscheint, soll ebenfalls über den Drucker ausgegeben werden. Diese Routine soll durch einen einfachen Tastaturcode aktivierbar und ebenso abschaltbar sein. Weil der Code dazu natürlich ständig im Speicher stehen muß, interessiert hier die Kürze besonders. Es gewinnt, wer diese Aufgabe am besten löst. Das Listing wird dann veröffentlicht, und der Sieger erhält zusätzlich zum Honorar einen Hunderter extra. Schickt die Programme mit kommentiertem Quellcode und Beschreibung bis zum 15. 6. 1991 an

Markt & Technik Verlag AG
64'er-Redaktion
Stichwort: Mitmachen — mitgewinnen
Hans-Pinsel-Straße 2
8013 Haar

Eigentlich klang die Aufgabe, die wir Ihnen in Ausgabe 1/90 gestellt hatten, recht einfach: eine Routine zum Zeichnen von Linien auf dem Grafikbildschirm des C64. Da dies jedoch möglichst schnell geschehen sollte, mußte man sich schon etwas einfallen lassen. Wie meist galt dann auch hier die Regel, je schneller ein Programm ist, um so länger wird es auch. Nichtsdestotrotz veröffentlichen wir wegen der interessanten Problemlösungen das ganze Siegerlisting. Es stammt von Gerhard Wesp aus Mattsee in Österreich.



Schnell Linien zeichnen – der Bresenham-Algorithmus macht's möglich

Ursprünglich wurde dieses Programm für einen C128 geschrieben, konnte dann jedoch auch für den C64 angepaßt werden. Die Routine basiert auf dem Bresenham-Algorithmus (beschrieben in Ausgabe 7/87, Seite 109).

Die meisten gängigen Linienalgorithmen berechnen für jeden einzelnen Punkt der zu zeichnenden Linie erneut Bildschirmadresse und Bit-Maske. Die Hauptidee dieser Routine ist jedoch, diese Daten rekursiv aus Bildschirmadresse und Bit-Maske des vorhergehenden Punktes zu ermitteln, was einen stark verminderten Rechen- und damit Zeitaufwand mit sich bringt.

Dazu ein Beispiel: Der erste Punkt einer Linie habe die Koordinaten 2/4. Seine Bildschirmadresse (bezogen auf die Bitmap ab $2000) ist $2004, seine Bit-Maske %0010 0000 ist $20. Der nächste Punkt sei um ein Pixel rechts und unten davon zu setzen. Die Bildschirmadresse des zweiten Punktes ergibt sich dann einfach, wenn man die des ersten um 1 erhöht, und die Bit-Maske, wenn man sie um eine Stelle nach rechts verschiebt (mit dem Assembler-Befehl LSR).

Leider ist dies jedoch nicht immer so einfach: Durch den komplizierten Aufbau der Bitmap-Grafik beim C64 kann, wenn die y-Koordinate eines Punktes z.B. 7 ist, nicht einfach die Bildschirmadresse inkrementiert werden. Die y-Koordinate wäre dann wieder 0, dafür aber x um eins erhöht. Um von der siebten in die achte oder allgemein von der 8n-1. auf die 8n. Zeile zu kommen, muß man zur Bildschirmadresse 313 addieren. Der Algorithmus prüft also bei jeder Erhöhung des y-Wertes, ob er gerade in einer 8n-1. Zeile ist. Dieses Problem wurde hier mit einer Tabelle gelöst, die für jeden Wert des lower Bytes der Bildschirmkoordinate angibt, was addiert werden muß, um in die nächste Zeile zu gelangen.

Die Entscheidung, welche Koordinaten der nächste Punkt hat (also ob die x- oder y-Koordinate oder beide inkrementiert werden), trifft der Bresenham-Algirithmus.

Als weitere geschwindigkeitssteigernde Maßnahme wurde die Technik der Selbstmodifikation verwendet (bei den beiden Additionskonstanten I1 und I2), wie in dem Listing zu sehen ist.

Auf der Programmservicediskette zu dieser Ausgabe befindet sich neben dem Quellcode auch noch ein Demoprogramm, das einige Möglichkeiten der Routine zeigt. (hb)

Listing »Fastline«, das Siegerprogramm zu »Mitmachen -- mitgewinnen«

```
READY.
0     -.SYNTAX
1     -.NONUM
2     -.LIST 1,3
3     -.SKIP 255,0
4     -.SYMBOLS W,1,3
100   -                       .DEFINE X= $FA  ;KOORDINATEN DES
110   -                       .DEFINE Y= X+2  ;STARTPUNKTS
120   -                       .DEFINE XZ= $A3 ;UND
130   -                       .DEFINE YZ= XZ+2;DES ZIELPUNKTS
140   -                       .DEFINE DX= $57 ;BRESENHAM-
150   -                       .DEFINE DY= DX+2;VARIABLEN
160   -                       .DEFINE I2= DY+2;BS-ADRESSE
170   -                       .DEFINE AD= I2+2;MOMENT. PUNKT
180   -                       .DEFINE D= AD+2 ;ZAEHLER NOCH
190   -                       .DEFINE C= D+2  ;VERBLEIBENDE PUNKTE
200   -                       .DEFINE BY= C+2 ;BITMASKE
210   -                       .DEFINE INCTABL= $C300;TABELLEN FUER
220   -                       .DEFINE INCTABH= INCTABL+256;LINE-ROUTINE
230   -                       .DEFINE LOTAB= INCTABH+256;TABELLE
240   -                       .DEFINE HITAB= LOTAB+200;BILDSCHIRMADRESSEN
250   -.BASE $C000
260   -                       JSR GETCOR      ;KOORDINATEN HOLEN
270   -                       SEC
280   -                       LDA YZ          ;
290   -                       SBC Y           ;DY = YZ - Y
300   -                       STA DY          ;
310   -                       BCS PIXEL
320   -;
330   -                       EOR #$FF        ;WENN DY < 0
340   -                       ADC #1          ;DANN
350   -                       STA DY          ;DY = ABS(DY)
360   -;
370   -                       LDA Y           ;START- UND
380   -                       LDX YZ          ;ZIELPUNKTE
390   -                       STA YZ          ;VERTAUSCHEN
400   -                       STX Y           ;SODASS IMMER
410   -;
420   -                       .LAX X           ;VON OBEN
430   -                       LDY XZ          ;NACH UNTEN
440   -                       STY X           ;GEZEICHNET
450   -                       LDY XZ+1        ;WIRD
460   -                       STY X+1
470   -                       .SAX XZ
480   -;
490   -PIXEL                  LDA #0          ;DA Y < 200
500   -                       STA DY+1        ;MUSS HBYTE 0 SEIN
510   -;
520   -                       LDA X           ;BS-ADRESSE DES
530   -                       AND #$F8        ;STARTPUNKTES
540   -                       LDY Y           ;BERECHNEN
550   -                       CLC             ;UND
560   -                       ADC LOTAB,Y     ;IN
570   -                       STA AD          ;AD
580   -                       LDA X+1         ;SPEICHERN
590   -                       ADC HITAB,Y
600   -                       STA AD+1
610   -;
620   -                       LDA X           ;BITMASKE
630   -                       AND #7          ;DES
640   -                       TAX             ;STARTPUNKTS
650   -                       LDA BYTAB,X     ;NACH
660   -                       STA BY          ;BY
670   -;
680   -                       SEC
690   -                       LDA XZ
700   -                       SBC X
710   -                       STA DX
720   -                       LDA XZ+1        ;DX = XZ - X
730   -                       SBC X+1
740   -                       STA DX+1
750   -                       PHP             ;DX < 0?
760   -                       BCS DXDYCOMP
770   -;
780   -                       EOR #$FF        ;WENN
790   -                       STA DX+1        ;DX < 0
800   -                       LDA DX
810   -                       EOR #$FF        ;DANN
820   -                       ADC #1
830   -                       STA DX
840   -                       BCC DXDYCOMP    ;DX = ABS(DX)
850   -                       INC DX+1
860   -;
870   -DXDYCOMP               CLC             ;DX
880   -                       LDA DX+1        ;MIT
890   -                       BNE DXDYSWAP    ;DY
900   -                       LDA DY          ;VERGLEICHEN
910   -                       CMP DX
920   -;
930   -DXDYSWAP               PHP             ;ERGEBNIS MERKEN
940   -                       BCC SETCOUNT    ;SPRUNG,WENN DX>DY
950   -                       .LAX DX          ;DX
960   -                       LDY DY          ;UND
970   -                       STY DX          ;DY
980   -                       .SAX DY
990   -                       LDA #0
1000  -                       STA DX+1
1010  -;
1020  -SETCOUNT               LDA DX          ;STARTWERT FUER
1030  -                       EOR #$FF        ;COUNTER SETZEN
1040  -                       STA C           ;WIRD IM LINIEN-
1050  -                       LDA DX+1        ;ALGORITHMUS
1060  -                       EOR #$FF        ;INKREMENTIERT BIS
1070  -                       STA C+1         ;ER 0 IST
1080  -;
1090  -                       ASL DY          ;DY = 2 * DY
1100  -                       ROL DY+1
1110  -;
1120  -                       SEC             ;D = DY - DX
1130  -                       LDA DY          ;BEACHTEN, DASS
1140  -                       SBC DX          ;DY VERDOPPELT
1150  -                       STA D           ;WURDE
1160  -                       LDA DY+1
1170  -                       SBC DX+1
1180  -                       STA D+1
1190  -;
1200  -                       SEC             ;I2 =
1210  -                       LDA D           ;D - DX
1220  -                       SBC DX          ;(= DY - 2DX)
1230  -                       STA I2
1240  -                       LDA D+1
1250  -                       SBC DX+1
1260  -                       STA I2+1
1270  -;
1280  -                       PLP             ;VERGLEICHSERG
1290  -                       BCC LINE1       ;ZURUECKHOLEN
1300  -                       PLP
1310  -                       LDY #0
1320  -                       JSR XINCMOD
1330  -                       .LAX DY          ;ADDITIONSKONSTANTEN
1340  -                       STA I1L2+1      ;I1 UND I2
1350  -                       STX I1H2+1      ;DURCH SELBST-
1360  -                       .LAX I2          ;MODIFIKATION
1370  -                       STA I2L2+1      ;FESTSETZEN
1380  -                       STX I2H2+1
1390  -                       JMP PLOT2       ;LINIE ZEICHNEN
1400  -;
1410  -LINE1                  PLP             ;ANALOG
1420  -                       LDY #(XINC1-XINC2)
1430  -                       JSR XINCMOD
1440  -                       .LAX DY
1450  -                       STA I1L1+1
1460  -                       STX I1H1+1      ;ZU
1470  -                       .LAX I2
1480  -                       STA I2L1+1
1490  -                       STX I2H1+1      ;OBEN
1500  -                       JMP PLOT1
1510  -;
1520  -XINCMOD                LDA #0          ;WENN CARRY GESETZT
1530  -                       ADC #0          ;IST X = 1
1540  -                       TAX
1550  -                       LDA MODTAB,X
1560  -                       STA XINC2,Y
1570  -                       LDA MODTAB+2,X
1580  -                       STA XINC2+4,Y
1590  -                       LDA MODTAB+4,X
1600  -                       STA XINC2+8,Y
1610  -                       LDA MODTAB+6,X
1620  -                       STA XINC2+10,Y
1630  -                       LDA MODTAB+8,X
1640  -                       STA XINC2+12,Y
1650  -;
1660  -                       LDY AD          ;LBYTE DER ADRESSE
1670  -                       LDA #0          ;INS
1680  -                       STA AD          ;Y-REGISTER
1690  -                       RTS
1700  -;
1710  -I2ADD2                 LDA D           ;ALGORITHMUS
1720  -I2L2                   ADC #$FF        ;FUER
1730  -                       STA D           ;DY >= DX
1740  -                       LDA D+1
1750  -I2H2                   ADC #$FF
1760  -                       STA D+1
1770  -;
1780  -XINC2                  ASL BY
1790  -                       BCC YINC2
1800  -                       ROL BY
1810  -                       TYA
1820  -                       ADC #$F8
1830  -                       TAY
1840  -                       BCS YINC2
1850  -                       DEC AD+1
1860  -;
1870  -YINC2                  TYA
1880  -                       TAX
1890  -                       LDA INCTABL,X
1900  -                       TAY
1910  -                       AND #7
1920  -                       BNE PLOT2
1930  -                       CLC
1940  -                       LDA INCTABH,X
1950  -                       ADC AD+1
1960  -                       STA AD+1
1970  -;
1980  -PLOT2                  CLC             ;PUNKT SETZEN
1990  -                       LDA (AD),Y      ;CARRY FUER
2000  -                       ORA BY          ;SPAETERE ADDITIONEN
2010  -                       STA (AD),Y      ;LOESCHEN
2020  -;
2030  -                       INC C           ;ZAEHLER ERHOEHEN
2040  -                       BEQ END         ;ENDE, WENN 0 ERREICHT
2050  -;
2060  -                       BIT D+1
2070  -                       BPL I2ADD2
2080  -;
2090  -                       LDA D
2100  -I1L2                   ADC #$FF
2110  -                       STA D
2120  -                       LDA D+1
2130  -I1H2                   ADC #$FF
2140  -                       STA D+1
2150  -                       JMP YINC2
2160  -;
2170  -END                    RTS
2180  -;
2190  -I2ADD1                 LDA D           ;ALGORITHMUS FUER
2200  -I2L1                   ADC #$FF        ;DY < DX
2210  -                       STA D
2220  -                       LDA D+1
2230  -I2H1                   ADC #$FF
```

```
2240 -                  STA D+1
2250 -;
2260 -                  TYA
2270 -                  TAX
2280 -                  LDA INCTABL,X
2290 -                  TAY
2300 -                  AND #7
2310 -                  BNE XINC1
2320 -                  CLC
2330 -                  LDA INCTABH,X
2340 -                  ADC AD+1
2350 -                  STA AD+1
2360 -;
2370 -XINC1             LSR BY
2380 -                  BCC PLOT1
2390 -                  ROR BY
2400 -                  TYA
2410 -                  ADC #8
2420 -                  TAY
2430 -                  BCC PLOT1
2440 -                  INC AD+1
2450 -;
2460 -PLOT1             CLC
2470 -                  LDA (AD),Y
2480 -                  ORA BY
2490 -                  STA (AD),Y
2500 -;
2510 -                  INC C
2520 -                  BNE DCOMP1
2530 -                  INC C+1
2540 -                  BEQ END
2550 -;
2560 -DCOMP1            BIT D+1
2570 -                  BPL I2ADD1
2580 -;
2590 -                  LDA D
2600 -I1L1              ADC #$FF
2610 -                  STA D
2620 -                  LDA D+1
2630 -I1H1              ADC #$FF
2640 -                  STA D+1
2650 -                  JMP XINC1
2660 -;
2670 -GETCOR            JSR CHKCOR
2680 -                  STX Y
2690 -                  .LAX $14
2700 -                  .SAX X
2710 -                  JSR CHKCOR
2720 -                  STX YZ
2730 -                  .LAX $14
2740 -                  .SAX XZ
2750 -                  RTS
2760 -;
2770 -CHKCOR            JSR $AEFD       ;CHKCOM
2780 -                  JSR $B7EB       ;GETADR UND GETBYTE
2790 -                  CPX #200        ;Y-KOORD. PRUEFEN
2800 -                  BCS ILLQUANT
2810 -                  LDA $14         ;X-KOORD. PRUEFEN
2820 -                  CMP #<(320)
2830 -                  LDA $15
2840 -                  SBC #>(320)
2850 -                  BCS ILLQUANT
2860 -                  RTS
2870 -;
2880 -ILLQUANT          JSR AUS
2890 -                  JMP $B248
2900 -;
2910 -                  JSR $B7F1
2920 -                  CPX #0
2930 -                  BEQ AUS
2940 -                  LDA $D011
2950 -                  ORA #$20
2960 -                  STA $D011
2970 -                  LDA #$38
2980 -                  STA $D018
2990 -                  RTS
3000 -;
3010 -AUS               LDA $D011
3020 -                  AND #$DF
3030 -                  STA $D011
3040 -                  LDA #$14
3050 -                  STA $D018
3060 -                  RTS
3070 -;
3080 -                  JSR $B7F1
3090 -                  TXA
3100 -                  LDY #0
3110 -                  STY C
3120 -                  LDX #$C
3130 -                  STX C+1
3140 -                  LDX #4
3150 -                  JSR LOOP
3160 -                  LDX #$20
3170 -                  STX C+1
3180 -                  LDA #0
3190 -LOOP              STA (C),Y
3200 -                  INY
3210 -                  BNE LOOP
3220 -                  INC C+1
3230 -                  DEX
3240 -                  BNE LOOP
3250 -                  RTS
3260 -BYTAB             .BYTE $80,$40,$20,$10,8,4,2,1
3270 -MODTAB            .BYTE $06,$46,$26,$66,$F8,$08,$B0,$90,$C6,$E6
```


Grundlagen Floppyprogrammierung

Relativ schnell

Relative Dateien vereinfachen und beschleunigen die Suche nach bestimmten Daten ganz wesentlich. Wir zeigen Ihnen, wie man dies programmiert.

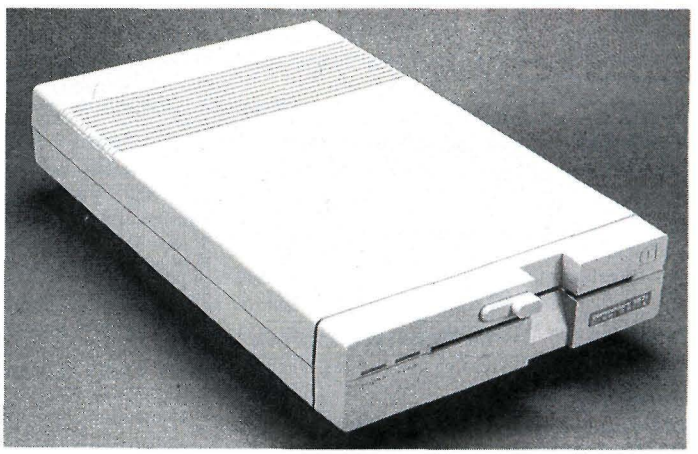
von Heinz Behling

Nachdem in Folge 1 schon die ersten Daten in einer sequentiellen Datei gespeichert wurden, wenden wir uns den wesentlich komfortableren relativen Dateien zu. Diese vermeiden nämlich den großen Nachteil des rein sequentiellen Zugriffs, d. h., Sie müssen nicht erst 100 Datensätze von der Diskette lesen, um den 101. in den Speicher zu bekommen. Mit besonderen Befehlen ist jeder beliebige Satz, ja sogar jedes einzelne Byte direkt zu erreichen. Dies Verfahren funktioniert ähnlich wie bei einer Schallplatte, bei der Sie ja durch entsprechendes Positionieren des Tonarms auch jeden Titel erreichen können. Daß auf diese Weise die Zugriffszeit wesentlich reduziert wird, dürfte klar sein.

Doch nun zu den Befehlen: Zunächst muß auch eine relative Datei eröffnet werden.

```
OPEN 2,8,2, "Reldatei,L" + CHR$(Satzlänge)
```

In diesem OPEN-Befehl steht einiges, was Ihnen sicherlich noch neu erscheinen wird. Das erste dürfte das »L« hinter dem Dateinamen sein. Dies ist ähnlich wie das »S«, das bei sequentiellen Dateien verwendet wurde. Es handelt sich um das Kennzeichen für relative Dateien. Als nächste, sehr wichtige Angabe folgt die Datensatzlänge. Hierbei ist es wichtig zu wissen, daß bei diesem Dateityp alle Sätze die gleiche Länge haben müssen. Sonst wäre es für die Floppy äußerst schwierig, die Position eines einzelnen Satzes zu bestimmen. Da dieser Wert schon von vornherein feststehen muß, sollten Sie sich genau überlegen, wie viele Sie in einem Datensatz brauchen.



Floppylaufwerke zu programmieren ist nicht immer einfach. Mit unserer Einführung liefern wir das nötige Rüstzeug.

Um nun auf die richtige Stelle in solch einer Datei zugreifen zu können (zu positionieren), muß man eben die Nummer des gewünschten Satzes der Floppy mitteilen können. Wie schicken wir