

Det eneste blad med programmer udelukkende til Commodore

# LISTEN



Marts / April 1988

# 1

Kun Dkr. 28,50

Nok. 28,50

Korte og lange, seriøse og spil-programmer

Hele 40 tætpakkede sider

Commodore 64 og 128

C-16

Tonsvis af programmer

Masser af programmer til Commodore 64 og 128

# INPUT

## VELKOMMEN

Vi byder alle velkommen til vort nye indtaste-magasin! Det er faktisk det eneste programblad, der findes, idet bladene »TAST« og »INPUT« programbladene lukkede, mens dette blad var under udarbejdelse.

Vi bringer udelukkende COMMODORE-programmer, i modsætning til »TAST«, der bragte programmer til alle mulige og umulige computere (Newbrain, Marathon osv.). Der var vel ingen, der fik fuld valuta for prisen, der var på hele 34,85 kr.

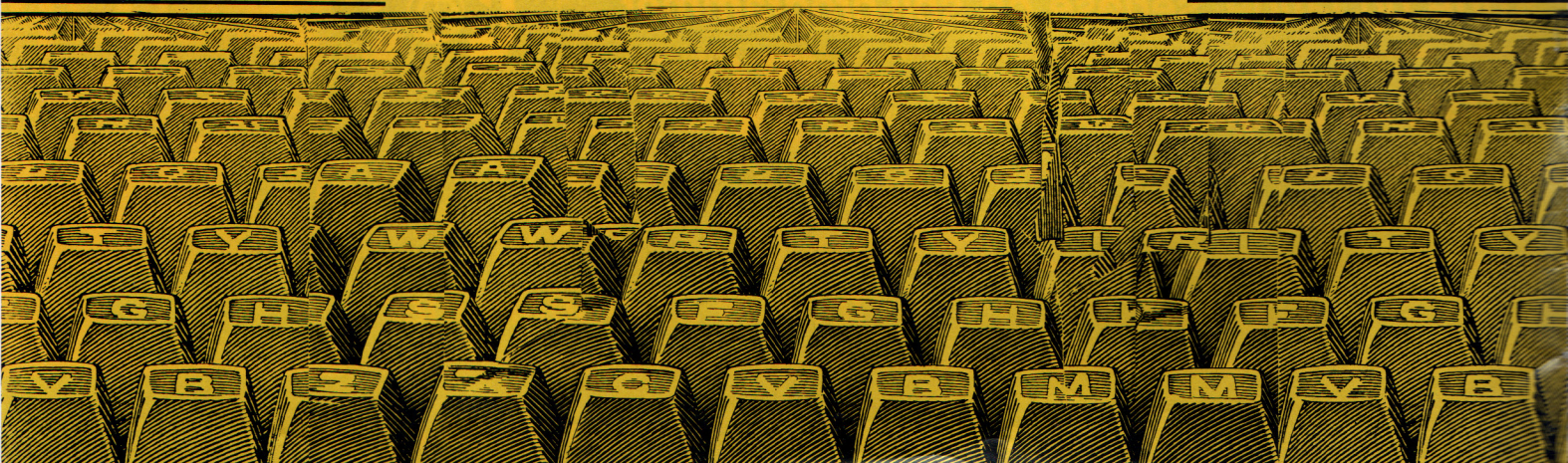
»INPUT« bragte kun AMSTRAD-programmer og det til en rimelig pris, men desværre for »INPUT« har COMMODORE nu overtaget hele hjemmecomputermarkedet rub og stub.

Det var med andre ord på tale at der kom et rent COMMODORE-programblad. Vi har lagt stor vægt på at fordele programmerne i forhold til fordelingen af Commodore-brugerne, dvs. at der er masser af C-64 og 128 programmer. Vor særdeles dygtige programredaktør Bo R. Petersen har selv lavet ca. halvdelen af programmerne, mens resten er udvalgt af de bedste indsendte læserprogrammer.

Disse er næsten alle blevet bearbejdet og komprimeret så indtastearbejdet er reduceret, og vi undgår at lave et blad på 100 sider. Alle programmer er forsynet med en fejlfinder, så du kan finde de næsten uundgåelige taste-fejl. De længere programmer er opdelt i intervaller, så du ikke skal lede hele programmet igennem for at finde fejlen. Alligevel må man påregne en vis tid til fejlretning. Alle programmerne er afprøvede og fri for fejl.

Næste nummer af bladet kommer først, når vi igen har samlet programmer, der er gode nok.

**GOD FORNØJELSE MED INDTASTNINGEN**





# INDHOLD

**Aventurelistning:**

<i>Din vej til stjernerne</i> . . . . .	4
<b>Vinderprogrammet fra vor AMIGA-konkurrence:</b>	
<i>Intruder C64/128</i> . . . . .	8
<i>128 programmer</i> . . . . .	11
<i>Danske Tegn/Dannebrog C64/128</i> . . . . .	13
<i>Hot Space Duel C64/128</i> . . . . .	14
<i>4 på stribe C64/128</i> . . . . .	16
<i>Sprite Bouncer C64/128</i> . . . . .	18
<i>Lock Keys C64/128</i> . . . . .	19
<i>Editor C64/128</i> . . . . .	20
<i>Professor Terning/F-Tools C64/128</i> . . . . .	22
<i>Erazor/Båndsalat C64/128</i> . . . . .	24
<i>OK/Reset/Game Chars C64/128</i> . . . . .	26
<i>Verity/Disco Lights C64/128</i> . . . . .	28
<i>CAR RACE/Super Old C-16</i> . . . . .	29
<i>Sådan laver du dit eget spil</i> . . . . .	35

**Chefredaktør og ansvarlig udgiver:**  
**Hans Chr. Thaysen / IC RUN**

**Programredaktør: Bo R. Petersen**



**Din Vej Til Stjerne...**

# LIST ADVENTURESPILE SYSTEM



...eller ned i mørke grotter. Din vej til at lave hurtige og store adventurespil i basic. Det lyder utroligt, men med LIST ved hånden er det muligt at lave adventurespil, der i kompleksitet og kvalitet langt overgår ting man kan købe for 200 kr. i handelen. Her ser du en listning af et lille adventurespil. Det er selvfølgelig ikke meningen at du skal taste det ind for at spille det. Det ville være kedeligt, du havde jo set alle hemmelighederne mens du tastede det ind. Men spillet er bygget på et system, som du kan bruge selv. At det er et spil og ikke bare et system skyldtes, at jeg syntes at det er den bedste måde at forklare tingene på.

Men forøvrigt så, skriv da bare programmet ind. Sølle 8K. Og spil det igennem, så du er sikker på at kunne følge med i resten af artiklen.

Nu er det jo sådan at kernen i et adventurespil er tekst, tonsvis af tekst. Tekst der beskriver, tekst der forklarer og tekst der fylder. Og imellem al sludderet er der nogle betingelser, der bestemmer hvornår teksten skal spyttes ud. I nogle spil er der også billeder. De fylder mere end tekst og er i grunden kun for folk med lav fantasi.

Adventurespil består i sin dybere substans af nogle få grundpiller.

**STEDER:** Det er her det hele foregår. Om det er i grotter, på rumskibe eller oppe i skolen.

**LØSE TING:** Det er alle de ting, som folk har tabt rundt omkring. Det er alle de ting du tager, smider og bruger.

**FASTE TING:** Det er alle de ting, som ikke bare er en del af stedsbeskrivelsen, men alligevel ikke kan samles op. Det er monstre i huler, og vagtbotter på rumkolonien.

Når man har fået organise-

ret sit spil sådan er det meget nemmere at holde styr på og lagre tingene.

Hermed følger en systematisk gennemgang af programmet, linie for linie, rutine for rutine.

**STEDER** ligger med en lille rutine pr. lokation. Den skriver beskrivelser ud og bestemmer i hvilke retninger man kan gå og hvor man kommer hen. Dette bestemmes med numre fra 3 og opad. De to første steder (1&2) er 1: et sted man smider ting der ikke har andre steder at være. 2: din lomme.

**LØSE TING** er defineret ved hjælp af en variabel og tre tekster, ud over at de også har et nummer. Variablen er det sted hvor tingen er. Så er der det korte navn for tingen (eks. BOG), det gængse navn (eks. EN BOG) og beskrivelsen (eks. EN TYK LÆDER-INDBUNDET BOG).

**FASTE TING** opstår alle mulige steder i programmet og da det er så løst et begreb kan man ikke sætte det på DATA-linier. Men som regel er der knyttet et eller flere flag til dem. De kan være sat eller ikke sat. Tingen kan være tilfredsstillet eller ej. En dør kan være åben eller lukket.

Da du spillede det listede adventure, og det har du selvfølgelig gjort, nikkede du nok genkendende til noget. Det var velsagtens stederne. De tre positioner er nemlig de tre første fra verdens ældste adventurespil, nemlig Colossal Adventure. Men da der skulle være noget spil i det, ændrede jeg originalen. Jeg håber ikke at dem, der ejer rettingerne har noget imod det.

Med udgangspunkt i spillet beskrives rutinerne.

Det første du lagde mærke



til var vel uhyret. Det er en ting, der bliver på den position, ligegyldigt hvad du gør (og gudske-tak-og-lov for det), men det betyder altså at det er en **FAST TING**. Vel at mærke en aktiv **FAST TING**, så den bliver kaldt fra 200 og handlingen foregår så i 2500-. HT er det man kunne kalde en **Handlings Tæller**, den tæller en frem hver gang rutinen bliver kaldt og uhyret foretager sig altså noget nyt, hver gang man står uden for huset.

Det næste du ser er sværdet. Sværdet er defineret i linie 8000. Her kan vi se at sværdet begynder i position 3 og de tre tekster der følger med tingen.

Det første man kunne finde på, at gøre med sådant et sværd, er jo nok at samle det op.

Dette sker i linierne 1200-. Hvis det andet ord er blevet genkendt som en ting, checker programmet om denne ting ligger på det samme sted som man selv er. Hvis man har mindre end tre ting i forvejen, tager man så tingen. Nu er tingen et sværd. Den

mest naturlige reaktion på nu at have et våben er kun naturlig..Drabsafdelingen ligger fra linie 1800.

Nu er spillet så stille og fredeligt at det kun er uhyret man skal dræbe. Hvis det er det man vil er der to muligheder, enten at angribe før man tager sværdet eller at gøre det efter. Bagefter skal man ind i huset. Osv. Osv.

Læs programmet igen, og hvis der er ting du ikke forstår hjælper det at kigge i variabelisten.

Leg lidt med programmet og se at få lavet nogle spil. Hvis du får problemer, så skriv til IC/RUN. Dine spørgsmål vil nå de rette hænder før eller siden. Samme hænder glæder sig også meget til at se hvad du kan få ud af systemet. Og det er ikke småting, personligt har jeg lavet et 35 K's adventure med over hundrede rum og personer i. Hvis der viser sig interesse, kan der også komme udvidelser til systemet engang ud i fremtiden og færdige spil til at downloade fra Databasen.

Der er fremtid i LIST's Adventure System.

10- Variabler læses og tabeller udfyldes. Se listen sidst i teksten.

100- Spillet startes op.

200- Her sker ting. Rutinen til uhyret bliver kaldt. I det hele taget er det her at de fleste hændelser skal styres fra.

260- Et input læses og splittes op ord for ord. De tre første ord puttes i a\$, b\$ og c\$. Hvis det andet ord er en LØS TING sættes TF til denne tings nummer.

400- Her tester programmet dit input og sender videre til den aktuelle rutine.

1000- Bevægelse fra sted til sted foregår her. I linie 1010 og 1020 bliver der checket for hvorvidt man kan gå igennem døren eller ej.

1100- Her skrives stedbeskrivelser ud og ting der ligger på stedet nævnes.

1200- Ting tages.

1300- Ting lægges.

1400- Alle de ting man har i lommen skrives ud.

1500- Der checkes for om den nævnte ting er en LØS TING, ellers gæses videre til 2200. LØSE TING beskrives.

1600- Døren åbnes.

1700- Døren lukkes.

1800- Dræb ting. Uhyret foreksempel.

1900- Lås døren op eller i.

2000- Tænd lampen.

2100- Sluk lampen.

2200- Andre ting beskrives.

2500- Ting sker her.

3000- Spillet slutter.

3100- Scoren skrives ud.

3200- Du vinder.

8000- Tingenes data.

10000- Stedbeskrivelser.

```
10 rem ***** adventure-spil *****
20 rem
30 rem --variabler indstilles--
40 obj=4:lo=3:atb=1:po=0
50 dim s(obj),s$(obj),b$(obj),k$(obj),t$(3)
60 print chr$(147)
70 for i=1 to obj:read d1,d2$,d3$,d4$
80 s(i)=d1:k$(i)=d2$:s$(i)=d3$:b$(i)=d4$:next i
99 rem --programmet begynder--
100 print"Velkommen til LISTs adventure-system          Din vej til stjernerne.."
110 print
120 gosub 1100:rem stedbeskrivelse af det foerste rum
199 rem --grundrutine--
200 if lo=3 and uhyre=0 then gosub 2500
210 if lo=3 and s(2)=2 then gosub 4000
250 a$="":b$="":c$="":nr=1:print:traek=traek+1:tf=0
260 input "Hvad nu";t$
270 ls=len(t$)
280 for i=1 to ls
290 m$=mid$(t$,i,1)
300 if m$=" " then nr=nr+1:goto 320
310 t$(nr)=t$(nr)+m$
320 if nr>3 then 340
330 next i
340 a$=t$(1):b$=t$(2):c$=t$(3)
350 t$(1)="":t$(2)="":t$(3)="
360 for i=1 to obj
370 if b$=k$(i) then tf=i:goto 400
380 next i
399 rem --parser--
400 if a$="n" or a$="nord" then re=n:goto 1000
410 if a$="s" or a$="syd" then re=s:goto 1000
420 if a$="oe" or a$="oest" then re=oe:goto 1000
430 if a$="v" or a$="vest" then re=v:goto 1000
440 if a$="ind" then re=ind:goto 1000
450 if a$="ud" then re=ud:goto 1000
460 if a$="op" then re=op:goto 1000
470 if a$="ned" then re=ned:goto 1000
480 if a$="gaa" then a$=b$:goto 400
500 if a$="tag" then 1200
510 if a$="laeg" then 1300
520 if a$="inv" or a$="inventory" then 1400
530 if a$="beskriv" then 1500
540 if a$="aabn" then 1600
550 if a$="luk" then 1700
560 if a$="laas" then 1900
570 if a$="kig" then 1100
580 if a$="draeb" then 1800
590 if a$="flyv" then print "HOP... BOINK!! Jeg ramte jorden.":goto 200
600 if a$="point" then gosub 3100:goto 200
610 if a$="taend" then 2000
620 if a$="sluk" then 2100
630 if a$="quit" or a$="stop" then 3000
990 print "Det kan jeg ikke.":goto 200
999 rem --flytning--
1000 if re=0 then print "Den retning kan jeg ikke gaa.":goto 200
1010 if (lo<>3 or re<>4) and (lo<>4 or re<>3) then 1030
1020 if doer=0 or doer=1 then print "Doerren er lukket.":goto 200
1030 lo=re
1070 n=0:s=0:oe=0:v=0
1080 print "ok.":print:print
1090 lo=re
1099 rem --stedbeskrivelser skrives--
1100 on lo-2 goto 10000,10100,10200
1150 he=0
1160 for i=1 to obj:if s(i)<>lo then 1190
1170 if he=0 then print "Jeg kan se.."
1180 print s$(i):he=he+1
1190 next i
1195 goto 200
1199 rem --tag ting--
1200 if tf=0 then print "Den ting findes ikke.":goto 200
1210 if lo<>s(tf) then print "Den ting er ikke her.":goto 200
1220 if atb=3 then print "Jeg kan ikke baere mere.":goto 200
1230 if tf=2 and doer>0 and lo=5 then 2600
1240 s(tf)=2:atb=atb+1:print "ok.":goto 200
1299 rem --laeg ting--
1300 if tf=0 then print "Den ting findes ikke.":goto 200
1310 if s(tf)<>2 then print "Den ting har jeg ikke.":goto 200
1320 s(tf)=lo:atb=atb-1:print "ok.":goto 200
1399 rem --inventory--
1400 print "Jeg har..."
1410 if atb=0 then print "ingenting.":goto 200
1420 tal=atb
1430 for i=1 to obj
1440 if s(i)<>2 then 1470
1450 print s$(i):tal=tal-1
1460 if tal>0 then print ","
1470 if tal=0 then print ".":goto 200
1480 next i
1490 goto 200
1499 rem --beskriv ting--
1500 gosub 2200:rem test paa andre ting
1510 if s(tf)<>lo and s(tf)<>2 then print "Jeg kan ikke se nog";s$(tf):goto 200
1520 print b$(tf)
1530 goto 200
1599 rem --aabn--
1600 if b$<>"doer" then print "Lad nu det vaere.":goto 200
1610 if lo<>3 and lo<>4 then print "Du kan jo aabne for hjernen.":goto 200
1620 if doer=0 then print "Doeren vil ikke aabne.":goto 200
```

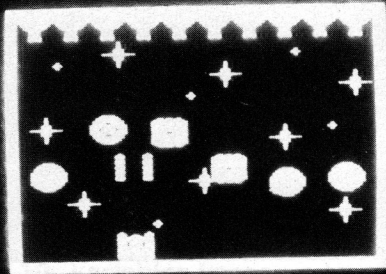
```

1630 if doer=2 then print "Doeren er allerede aaben.":goto 200
1640 print "kniirrkk... doeren aabner."
1650 doer=2:goto 200
1699 rem --luk--
1700 if c$="op" then 1600
1710 if lo<>3 and lo<>4 then print "Du kan jo aabne for hjernen.":goto 200
1720 if doer=1 or doer=0 then print "Doeren er allerede lukket.":goto 200
1730 print "smaekk!!"
1740 doer=1:goto 200
1799 rem --draeb--
1800 if b$<>"uhyre" then print "Det er vidst ikke noedvendigt.":goto 200
1810 if lo<>3 or uhyre=1 then print "Hva'for et uhyre?":goto 200
1820 if s(1)=2 then 1850
1830 print "Med et kampraab kaster du dig over"
1840 print "uhyret. Du blev";traek;" traek gammel.":goto 3000
1850 print "Med svaerdet hoejt haevet angriber du"
1860 print "uhyret. Fem svaerdhug og en frygtelig"
1870 print "masse sort blod senere!.."
1875 if p1=0 then p1=1:sco=sco+20
1880 print:print "Va'for et uhyre??!!":uhyre=1:goto 200
1899 rem --laas--
1900 if b$<>"doer" then print "Har du laast hjernen inde.":goto 200
1910 if lo<>3 and lo<>4 then print "Her er ikke nogen doer.":goto 200
1920 if doer=2 then print "Doerren er aaben":goto 200
1930 if s(4)<>2 then print "Du har ikke nogen noegle.":goto 200
1940 if c$="op" then 1970
1950 if doer=0 then print "Den er allerede laast!":goto 200
1960 print"Klikk!":doer=0:goto 200
1970 if doer=1 then print "Den er ikke laast.":goto 200
1975 if p2=0 then p2=1:sco=sco+10
1980 print"Klikk!":doer=1:goto 200
1999 rem --taend--
2000 if b$<>"lampe" then print "Pyroman!":goto 200
2010 if s(3)<>2 then print "Du har ikke nogen lampe":goto 200
2015 if p3=0 then p3=1:sco=sco+20
2020 print "Lampen er nu taendt":lys=1:goto 200
2100 if b$<>"lampe" then print "Lyseslukker!":goto 200
2110 if s(3)<>2 then print "Du har ikke nogen lampe":goto 200
2120 print "Lampen er nu slukket":lys=0:goto 200
2199 rem --beskrive andre ting--
2200 if b$<>"hus" or lo<>3 then 2210
2205 print "Et lille hus bygget af roede mursten.":goto 200
2210 if b$<>"uhyre" or lo<>3 or uhyre=1 then 2220
2215 print "Et slimet og ondsindet monster.":goto 200
2220 if b$<>"doer" or lo<>3 and lo<>4 then 2230
2222 if doer=0 then print "Den er laast.":goto 200
2225 if doer=1 then print "Den er lukket.":goto 200
2227 print "Den er aaben.":goto 200
2230 if b$<>"skov" or lo<>3 then 2240
2235 print "Moerk og fyldt med orker":goto 200
2240 rem flere beskrivelser
2296 print "Den ting kan du ikke se.":goto 200
2499 rem --leting der sker--
2500 rem --uhyret--
2510 tm=tm+1
2520 if tm=1 then print "Et uhyre kravler frem bag en sten."
2530 if tm=2 then print "Uhyret bevaeger sig taettere paa."
2540 if tm=3 then print "Uhyret slaar efter dig, men rammer ikke."
2550 if tm=4 then print "Uhyret rammer dig med en lige hoejre":goto 3000
2560 return
2599 rem --overfald--
2600 print "Pludselig kommer der larm oppefra."
2610 print "Fem orker vaelter ind i hytten og ned i"
2620 print "broenden, de tager skatten og er vaek"
2630 print "foer du kan naa at goere noget.":s(2)=1:goto 200
2999 rem --faerdig--
3000 print:print
3010 gosub 3100
3020 input "Vil du proeve igen (J/N)":jn$
3030 if jn$="n" then end
3040 run
3100 print "Du har loest";sco;"% indtil nu."
3110 return
3999 rem --vundet--
4000 print "Du har vundet... (gab)."
```

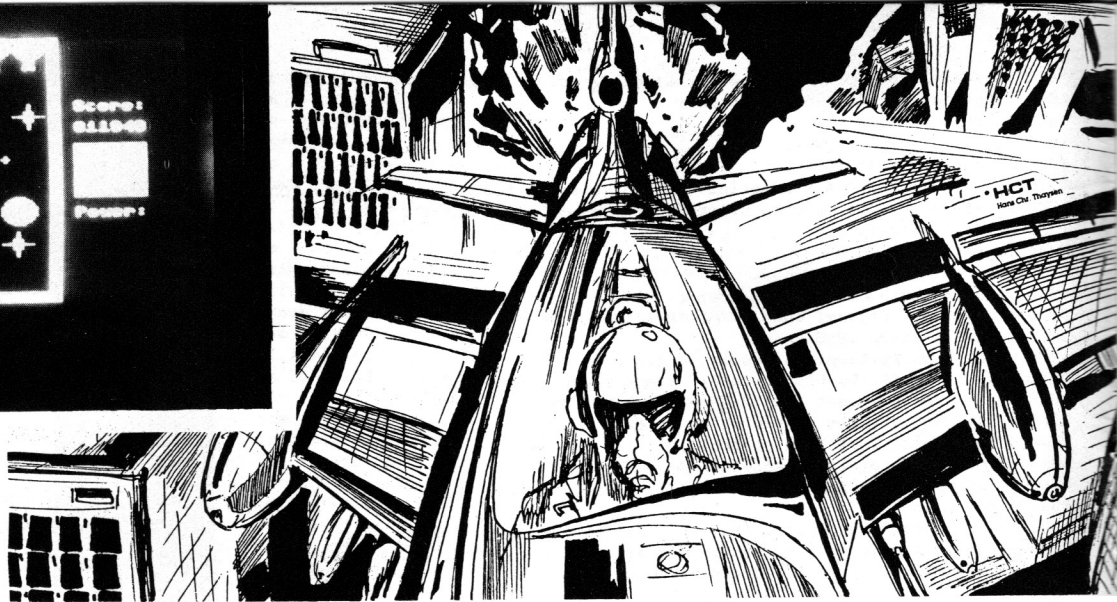
## Variabel Liste:

obj lo atb po s() k\$() b\$( ) doer uhyre  
A\$: Det 1. ord. Udsagnsordet.  
ATB: Antal Ting Båret  
B\$: Det 2. ord. Navneordet.  
B\$(x): Beskrivelsestekst af objekt x  
C\$: Det 3. ord. Et ekstra ord.  
DOER: Dørens tilstand 0-låst 1- lukket 2- åben  
K\$(x): Den korteste tekst af objekt x  
LO: Det nummer ens nuværende lokation har.  
LYS: Lampens tilstand 0-slukket 1-tændt (flag)  
PO: Point 1-100 procent  
RE: Nummeret på det sted man går imod.  
S(x): Nummeret på det sted hvor objekt x ligger.  
TF: Nummeret på den ting, som B\$ indeholder. 0 hvis B\$ ikke er en ting.  
TM: Tæller, der viser hvor langt uhyret er kommet.  
UHYRE: Uhyrets tilstand 0-levende 1-død





Score: 01104  
Power:



## Nummer et!

Program: The Intruders  
Gevinst: 1 stk. Amiga 500  
Udstyr: C-64/128

# INTRUDER

The Intruders er et lækkert shoot'em up spil i ren MC. Gameplayet er ganske simpelt, men virkelig underholdende. Sammen med den gode grafik og en god »feeling«, er dette spil uden tvivl, det bedste program vi har modtaget.

Du skal forhindre Wooberne (fjenderne) i, at komme ned under din y-akse. Det gør du simpelthen ved, at du hele tiden beskyder dem med din Pom-plaffer. Husk at ramme dem!! Undgå for alt i verden de små grimme Speccys, der, ustandseligt, bliver affyret fra nogle endnu grimmere kanoner!! Hvis du bliver ramt, mister du en h... masse power!!

Inden du går igang med at indtaste programmet, skal du skrive: poke 44,64:poke16384,0:new + RETURN

Når du har tastet programmet ind, run'er du det. Hvis der er nogle fejl i din indtastning, vil du blive gjort opmærksom på hvor fejlene er, og du kan derefter rette dem. Når programmet er fejlfrit, vil det på skærmen skrive: save»the intruders«,08

Cursoren vil stå oven på denne tekst, og hvis du kører med bånd, så ændrer du 8-tallet til et 1-tal. Derefter trykker du RETURN, og programmet vil blive gemt i en færdig version på henholdsvis disk eller bånd. Når dette er gjort, kan du bare loade den savede version og skrive RUN.

Programmet er indsendt af Christian Nielsen.

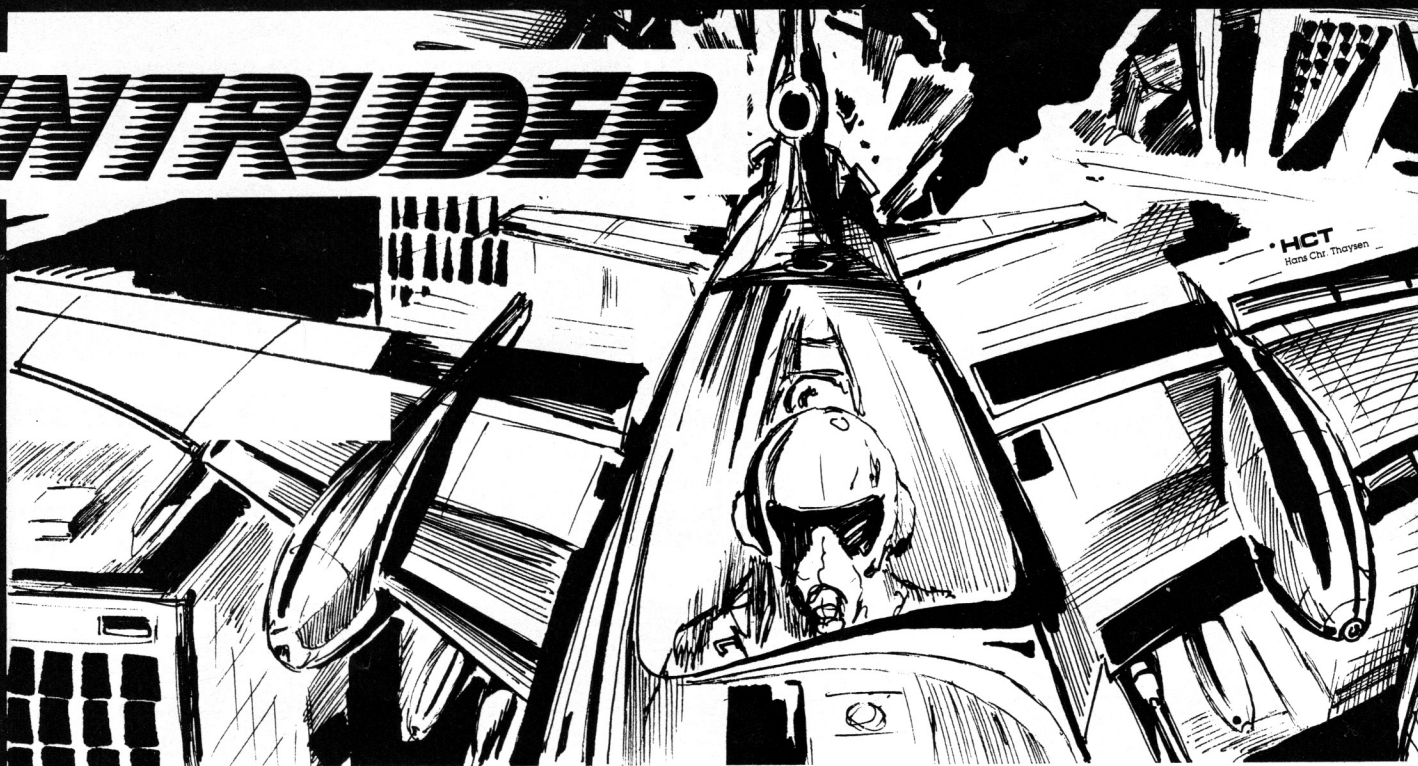
### The Intruders

```
100 data 29,8,195,7,158,50,48,56,48,32,67,72,82,73,83,84,73
110 data 65,78,32,78,73,69,76,83,69,78,0,0,0,0,120,169,128
120 data 141,17,208,160,0,140,32,208,140,16,208,140,21,208,140,23,208
130 data 140,29,208,132,171,140,18,208,32,248,9,169,51,133,1,185,0
140 data 220,153,0,208,185,0,221,153,0,209,185,0,222,153,0,210,136,8422
150 data 208,235,160,167,185,178,12,153,0,211,136,192,255,208,245,160,159
160 data 185,90,13,153,0,224,185,250,13,153,160,224,136,192,255,208,239
170 data 169,55,133,1,88,169,148,141,0,221,169,5,141,24,208,133,170
180 data 169,216,141,22,208,169,192,141,136,2,169,12,141,34,208,169,11
190 data 141,35,208,169,1,141,26,208,141,25,208,169,127,141,13,220,169,11801
200 data 157,141,20,3,169,12,141,21,3,169,47,141,40,15,169,154,160
210 data 14,32,30,171,169,27,141,17,208,173,1,220,201,239,208,249,120
220 data 169,128,141,17,208,169,145,141,20,3,169,12,141,21,3,169,15
230 data 141,33,208,141,37,208,169,8,141,134,2,32,68,229,160,33,162
240 data 6,32,12,229,169,41,160,15,32,30,171,160,0,169,103,153,1,8842
250 data 192,169,104,153,2,192,169,105,153,3,192,169,106,153,41,192,169
260 data 107,153,42,192,169,108,153,43,192,169,109,153,82,192,169,9,153
270 data 42,216,200,200,200,192,30,144,209,160,31,153,0,216,153,192,219
280 data 136,16,247,160,96,140,0,192,200,140,31,192,169,100,141,192,195,169
290 data 102,141,223,195,169,40,133,251,133,253,169,192,133,252,169,216,12303
300 data 133,254,162,22,160,0,169,98,145,251,169,9,145,253,160,31,169
310 data 99,145,251,169,9,145,253,165,251,24,105,40,133,251,133,253,144
320 data 4,230,252,230,254,202,16,218,169,101,160,29,153,193,195,136,16
330 data 250,169,255,141,21,208,141,28,208,160,7,185,95,15,153,39,208
```



340 data 185,103,15,153,248,195,136,16,241,160,15,185,79,15,153,0,208,11848  
350 data 136,16,247,169,6,141,38,208,162,7,189,111,15,133,252,189,119  
360 data 15,133,251,160,0,169,110,145,251,160,39,169,111,145,251,169,112  
370 data 200,145,251,169,113,200,145,251,169,114,160,80,145,251,189,127,15  
380 data 133,252,189,135,15,133,251,160,0,169,115,145,251,202,16,199,173  
390 data 30,208,169,27,141,17,208,76,65,12,162,5,169,0,149,172,157,11557  
400 data 76,3,202,16,246,96,198,170,16,43,169,20,133,170,165,168,73  
410 data 2,133,168,201,4,240,9,198,169,208,5,104,104,76,32,8,160  
420 data 5,153,177,217,153,217,217,153,1,218,153,41,218,153,81,218,136  
430 data 16,238,96,173,0,220,74,74,74,176,15,72,173,0,208,201,32  
440 data 240,6,56,233,2,141,0,208,104,74,176,15,72,173,0,208,201,10047  
450 data 248,240,6,24,105,2,141,0,208,104,74,176,17,165,171,208,13  
460 data 230,171,173,0,208,141,2,208,169,208,141,3,208,96,165,171,240  
470 data 23,173,3,208,56,233,8,141,3,208,201,72,176,10,169,0,133  
480 data 171,141,2,208,141,3,208,96,198,172,16,34,169,6,133,172,160  
490 data 11,162,5,185,4,208,24,105,1,201,250,208,5,169,29,32,49,9931  
500 data 11,153,4,208,157,64,3,202,136,136,16,230,96,173,30,208,74  
510 data 74,144,56,160,0,74,72,169,0,105,0,153,70,3,104,200,192  
520 data 6,208,241,160,11,162,5,189,70,3,240,25,189,64,3,201,200  
530 data 176,18,185,4,208,56,233,2,153,4,208,169,2,133,173,169,3  
540 data 32,34,12,202,136,136,16,221,165,168,201,20,144,64,164,174,240,9479  
550 data 60,165,169,208,9,173,0,208,201,234,176,28,144,47,173,0,208  
560 data 56,229,169,144,6,201,7,176,35,144,12,165,169,56,237,0,208  
570 data 144,25,201,23,176,21,165,175,145,251,169,8,145,253,32,201,11  
580 data 165,177,73,1,133,177,208,2,198,170,96,160,5,169,32,153,241  
590 data 194,169,0,153,241,218,136,16,243,164,170,48,14,169,116,153,241,10967  
600 data 194,169,15,153,241,218,136,16,243,96,104,104,32,121,12,160,3  
610 data 185,71,15,153,218,193,185,75,15,153,42,194,136,16,241,169,4  
620 data 133,168,169,32,133,169,169,0,133,170,76,98,12,165,174,240,51  
630 data 165,176,73,1,133,176,240,43,165,175,164,174,145,251,169,8,145  
640 data 253,165,251,24,105,40,133,251,133,253,144,4,230,252,230,254,201,11500  
650 data 192,240,15,177,251,133,175,169,116,145,251,169,15,145,253,230,168  
660 data 96,169,4,32,34,12,169,4,32,34,12,169,3,32,34,12,169  
670 data 0,133,174,96,165,174,208,59,173,0,208,56,233,12,74,74,74  
680 data 168,185,80,192,201,32,208,10,200,185,80,192,201,32,208,2,136  
690 data 136,132,174,169,0,133,168,152,10,10,10,105,24,133,169,169,80,9860  
700 data 133,251,133,253,169,192,133,252,169,216,133,254,177,251,133,175,96  
710 data 160,5,185,76,3,24,105,48,153,97,193,169,0,153,97,217,136  
720 data 16,239,96,133,181,152,72,138,72,166,181,254,76,3,189,76,3  
730 data 201,10,208,8,169,0,157,76,3,202,16,238,104,170,104,168,96  
740 data 32,137,12,32,53,10,32,115,10,32,143,10,32,206,11,32,130,9947  
750 data 11,32,182,10,32,107,12,32,14,12,32,60,11,76,65,12,32  
760 data 137,12,32,5,10,76,98,12,165,173,240,10,198,173,169,10,162  
770 data 8,160,9,208,6,169,15,162,12,160,11,141,33,208,142,34,208  
780 data 140,35,208,96,173,18,208,201,251,208,249,96,160,1,140,25,208  
790 data 136,140,18,208,76,49,234,172,40,15,185,248,14,141,33,208,206,8640  
800 data 40,15,16,230,169,47,141,40,15,208,223,170,85,64,127,112,113  
810 data 114,114,170,85,1,253,13,77,141,141,114,114,114,114,114,114,114  
820 data 114,141,141,141,141,141,141,141,141,114,114,113,112,127,64,85,170  
830 data 170,170,85,0,255,0,85,170,141,141,77,13,253,1,85,170,170  
840 data 85,0,255,3,83,147,144,170,85,0,255,255,255,255,255,170,85,10351  
850 data 0,255,192,197,198,6,148,164,165,233,233,250,249,254,255,60,0  
860 data 65,85,105,130,60,22,26,90,107,107,175,111,191,65,215,130,235  
870 data 150,65,0,40,255,255,239,239,223,239,223,223,255,255,255,164,255  
880 data 255,255,255,19,71,139,0,139,71,19,223,255,255,255,107,255,255,255  
890 data 255,223,239,223,239,239,255,255,255,255,239,223,71,223,239,255,14884  
900 data 255,170,150,65,60,60,65,150,170,48,0,12,220,0,55,236,60  
910 data 59,60,235,60,219,150,231,238,85,187,63,170,252,236,255,59,220  
920 data 235,55,236,150,59,60,235,60,236,60,59,223,235,247,217,150,103  
930 data 239,235,251,60,255,60,235,60,235,214,235,151,229,85,91,250,235  
940 data 175,63,60,252,153,48,0,12,236,0,59,152,0,38,236,0,59,11838  
950 data 48,0,12,236,0,59,152,0,38,236,0,59,48,0,12,236,0

# INTRUDER



```
960 data 59,152,0,38,236,0,59,48,0,12,236,0,59,152,0,38,236
970 data 0,59,48,0,12,236,0,59,152,0,38,236,0,59,0,0,0
980 data 33,0,255,0,3,150,192,14,85,176,13,190,112,55,235,220,62
990 data 170,188,58,150,172,250,90,175,233,90,171,233,106,171,229,106,171,7918
1000 data 229,170,171,233,170,171,250,170,175,58,170,172,62,170,188,55,235
1010 data 220,13,190,112,14,85,176,3,150,192,0,255,0,33,0,255,0
1020 data 3,150,192,14,105,176,13,190,112,58,255,172,55,195,220,59,130
1030 data 236,238,125,187,239,235,251,220,150,55,220,85,55,220,150,55,239
1040 data 235,251,238,125,187,59,130,236,55,195,220,58,255,172,13,190,112,12337
1050 data 14,105,176,3,150,192,0,255,0,33,15,60,240,57,235,108,230
1060 data 150,155,251,235,239,230,255,155,230,235,155,59,150,236,230,235,155
1070 data 230,255,155,59,195,236,230,0,155,59,195,236,230,255,155,230,235
1080 data 155,59,150,236,230,235,155,230,255,155,251,235,239,230,150,155,57
1090 data 235,108,15,60,240,33,144,147,8,212,72,69,32,201,78,84,82,13335
1100 data 85,68,69,82,83,33,13,45,45,45,45,45,45,45,45,45,45
1110 data 45,45,45,45,13,13,13,13,13,40,67,41,32,49,57,56,55
1120 data 32,66,89,13,13,195,72,82,73,83,84,73,65,78,32,206,73
1130 data 69,76,83,69,78,46,13,13,13,13,13,208,82,69,83,83,32
1140 data 211,208,193,195,197,32,84,79,32,80,76,65,89,33,0,0,5,5476
1150 data 5,3,3,3,13,13,13,13,15,15,15,15,15,10,10,10,10
1160 data 8,8,8,9,9,0,6,6,14,14,14,3,3,3,3,1,1
1170 data 1,1,1,15,15,15,15,12,12,12,11,11,0,144,211,67,79
1180 data 82,69,58,157,157,157,157,157,157,17,17,17,17,17,17,17,17
1190 data 17,17,208,79,87,69,82,58,0,71,1,13,5,79,22,5,18,3031
1200 data 140,220,0,0,40,75,80,80,120,85,160,85,200,80,240,75,4

1210 data 10,5,12,14,14,12,5,128,129,130,131,132,132,131,130,192,193
1220 data 194,194,194,193,192,193,129,147,236,104,174,13,218,207,192,193,195
1230 data 194,193,193,192,194,184,79,44,190,232,211,204,130,-1,8392
10000 clr:restore:adr=2049
10010 fort=1to23:chk=0:fore=1to85
10020 reada:ifa=-1then10040
10030 chk=chk+a:pokeadr,a:adr=adr+1:nexste
10040 reada:ifa<>chkthenprint"fejl i linjerne "50+(t*50)"-"90+(t*50):end
10050 nextt
10060 printchr$(147)chr$(18)" the intruders ":print:print
10070 print"lavet til "chr$(34)"list"chr$(34)" i 1987"
10080 print"af christian nielsen":print:print:print
10090 print"save "chr$(34)"the intruders"chr$(34)",8"
10100 printchr$(145)chr$(145)chr$(145);:poke43,1:poke44,8:poke45,143:poke46,15
```



### SPRITE HENTER

Her er et program der gør det muligt at bruge sprites overalt i computeren, der er demo indlagt.

```
10 FORT=0T039:READA$:POKE4864+T,DEC(A$):NEXT
20 DATAA2,00,A0,00,B1,FB,99,00
30 DATAOE,C8,D0,F8,E6,FC,B1,FB
40 DATA99,00,0F,C8,D0,F8,A2,00
50 DATAA9,38,9D,F8,07,18,69,01
60 DATAE8,E0,08,D0,F5,C6,FC,60
70 REM ***** DEMO *****
80 FORT=1T08:SPRITET,1,2:MOVSPRT.50+T*30,100:NEXT
90 FORT=0T065535STEP64
100 PRINT"(CLR)$";HEX$(T),T:S$=HEX$(T)
110 POKE252,DEC(LEFT$(S$,2))
120 POKE251,DEC(RIGHT$(S$,2))
130 SYS4864
140 GETKEYA$:NEXT
```

### COLOR

Dette program gør det muligt at ændre farverne på karaktererne. Syntaxen er »SYS 4864, FARVEKODE (0-15)«.

```
10 FORT=0T017:READA$:POKE4864+T,DEC(A$):NEXT
20 DATAA2,00,9D,00,D8,9D,00,D9
30 DATA9D,00,DA,9D,E8,DA,E8,D0
40 DATAF1,60
50 REM ***** DEMO *****
60 FORT=0T024:PRINTTAB(T)"IC RUN !!!!!":NEXT
70 FORT=1T015:SYS4864,T:FORY=0T0100:NEXTY,T:GOTO70
```

### RASTER HJÆLP

Her lidt hjælp til de mennesker der ikke har det så godt med rasterprogrammering. Der er demo i programmet.

```
10 FORT=0T082:READA$:POKE4864+T,DEC(A$):NEXT
20 DATA78,A9,21,8D,14,03,A9,13
30 DATA8D,15,03,A9,00,8D,12,D0
40 DATAA9,1B,8D,11,D0,A9,81,8D
50 DATA1A,D0,A9,7F,8D,0D,DC,58
60 DATA60,AD,19,D0,8D,19,D0,AD
70 DATA12,D0,CD,02,14,B0,12,AD
80 DATA03,14,8D,20,D0,8D,21,D0
90 DATAAD,02,14,8D,12,D0,4C,65
100 DATAFA,AD,01,14,8D,20,D0,8D
110 DATA21,D0,AD,00,14,8D,12,D0
120 DATA4C,75,FA
130 REM***** DEMO *****
140 SYS4864
150 FAST
160 A=5120:POKEA+1,00:POKEA+3,01
170 FORT=0T0230:POKEA,T:POKEA+2,T+15:NEXT
180 FORT=230T00STEP-1:POKEA,T:NEXT
190 FORT=245T015STEP-1:POKEA+2,T:NEXT
200 GOTO170
```

### 2 LINERS

Denne rutine laver en utrolig flot farve effect øverst på skærmen.

```
10 FORT=0T095:READA$:POKE4864+T,DEC(A$):NEXT
20 DATAAE,FF,03,BD,48,13,8D,00
30 DATAD8,8D,4F,D8,EA,E8,E0,18
40 DATAD0,02,A2,00,8E,FF,03,EA
50 DATAA2,26,BD,00,D8,9D,01,D8
60 DATAEA,E0,FF,D0,F5,EA,A2,00
70 DATABD,29,D8,9D,28,D8,E8,E0
80 DATA27,D0,F5,4C,65,FA,78,A9
90 DATA00,8D,14,03,A9,13,8D,15
100 DATA03,A9,00,8D,FF,03,58,60
110 DATA01,0F,0F,0C,0C,0C,0B,0B
120 DATA0B,0B,00,00,00,00,0B
130 DATA0B,0B,0B,0C,0C,0C,0F,0F
140 REM ***** DEMO *****
150 POKE53280,0:POKE53281,0
160 PRINT"(RVS)(CLR)(BLK) IC-RUN PRESENTERER ENDNU
EN NY COM-FECT ROUTINE"
170 SYS4918
```

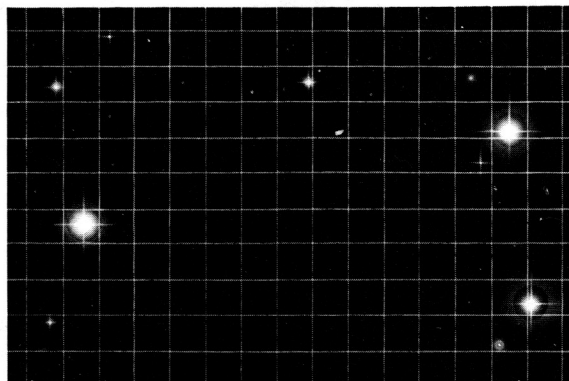
### DOBBEL PEEK

Denne rutine udskriver en side nul adresse ligesom »LDA (\$FB,X)«, der ændres adresse i linie 10.

```
10 X=DEC("314")
20 PRINTRIGHT$(HEX$(PEEK(X+1)),2);RIGHT$(HEX$(PEEK(X)),2)
```

# C 128

```
0 FORT=OTO95:READA$:POKE4864+T,DEC(A$):NEXT
10 DATA78,A9,00,85,FB,85,FD,A9
20 DATA20,85,FC,A9,80,85,FE,EA
30 DATAA2,00,A1,FB,18,01,FD,81
40 DATAFB,EA,E6,FB,E6,FD,A5,FD
50 DATAC9,00,D0,EC,E6,FC,E6,FE
60 DATAA5,FC,C9,40,D0,E2,58,60
70 DATAEA,EA,EA,78,A9,00,85,FB
80 DATA85,FD,A9,20,85,FC,A9,80
90 DATA85,FE,EA,A2,00,A1,FB,81
100 DATAFD,EA,E6,FB,E6,FD,A5,FD
110 DATAC9,00,D0,EF,E6,FC,E6,FE
120 DATAA5,FC,C9,40,D0,E5,58,60
125 REM***** DEMO *****
130 BANK 0
140 GRAPHIC1,1
150 FORT=OTO15
160 X=INT(RND(1)*321)
170 Y=INT(RND(1)*321)
180 Z=INT(RND(1)*321)
190 W=INT(RND(1)*321)
200 DRAW1,X,YTOZ,W
210 NEXT
220 CHAR1,0,0,"SCREEN I"
230 SLEEP1
240 SYS4915
250 GRAPHIC1,1
260 FORT=OTO15
270 X=INT(RND(1)*321)
280 Y=INT(RND(1)*321)
290 Z=INT(RND(1)*321)
300 W=INT(RND(1)*321)
310 DRAW1,X,YTOZ,W
320 NEXT
330 CHAR1,11,0,"SCREEN II"
340 SLEEP1
350 SYS4864:SLEEP1
360 CHAR1,9,0,"&"
```



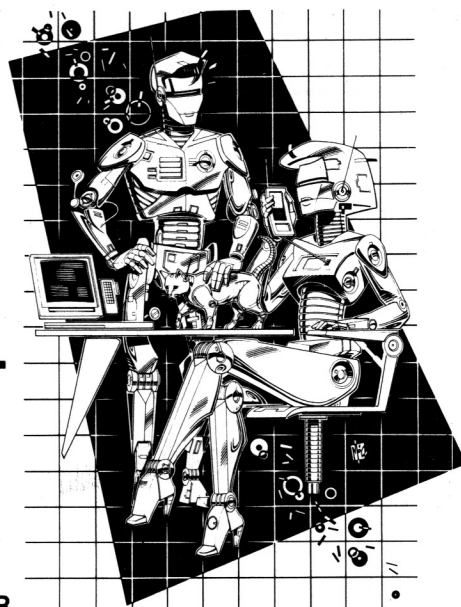
## SCREEN MIXER

Tro det eller lad vær, her er rutinen der får dig til at gå bag over af bare fryd. Programmet kan nemlig mixe to grafik skærme, for at mixe skal du gøre som følgende: Lav din tegning, skriv »SYS 4915«, for at flytte den, skriv »SYS 4864« for at mixe. Bare spænde sikkerhedsselen og kør dette program.

```
10 FORT=OTO49:READA$:POKE4864+T,DEC(A$):NEXT
20 DATAA9,00,85,FB,A9,04,85,FC
30 DATAA9,00,85,FD,A9,14,85,FE
40 DATAEA,A2,00,A1,FB,A8,A1,FD
50 DATA81,FB,98,81,FD,E6,FB,E6
60 DATAFD,A5,FB,C9,00,D0,EC,E6
70 DATAFC,E6,FE,A5,FC,C9,09,D0
80 DATAE2,60
90 REM***** DEMO *****
99 PRINT"(CLR)"
100 FORT=OTO24:CHAR1,T,T,"IC-RUN":NEXT
110 SYS4864
120 FORT=OTO24:CHAR1,T,T," ER BEDST":NEXT
130 SYS4864
135 SLEEP1
140 GOTO130
```

## 2 SCREENS

Med dette program har du mulighed for at bruge 2 tekst-skærme, du skifter mellem dem ved at skrive »SYS 4864«.



## SUPER FRYSER

Her er endnu et program der kan tryse en del af skærmen; det eneste du skal gøre er at skrive teksten i toppen af skærmen. Husk at afslutte teksten med en pil der peger mod venstre, herefter skrives »SYS 4864«.

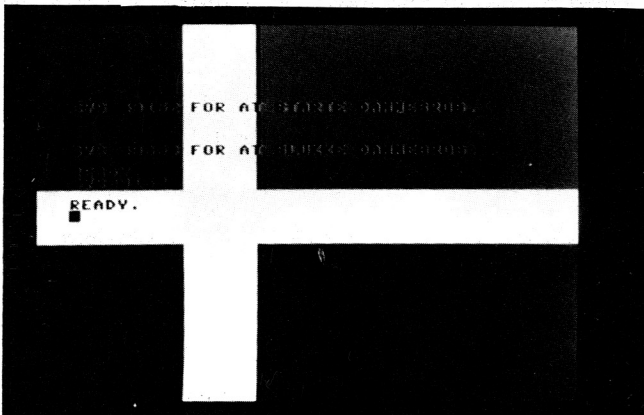
```
10 FORT=OTO41:READA$:POKE4864+T,DEC(A$):NEXT
20 DATA78,A2,00,BD,00,04,9D,00
30 DATA14,E8,D0,F7,A9,18,BD,14
40 DATA03,A9,13,8D,15,03,58,60
50 DATAA2,00,BD,00,14,C9,1F,F0
60 DATA06,9D,00,04,E8,D0,F3,4C
70 DATA65,FA
75 REM ***** DEMO *****
80 PRINT"(CLR)DIRECTORY←"
90 SYS4864
```

## Danske Tegn

Bruger man sin 64'er meget, bliver man hurtigt irriteret over, at der ikke findes danske bogstaver på trods af at man har købt den i Danmark! Dette har jeg her rådet bod på, for mig irriterer det sgu! Ikke nok med, at jeg har lavet danske karakterer, nææh, jeg har skam også udvidet Basic-ram'en med ca. 1.2 Kb ekstra. Det kan dog godt

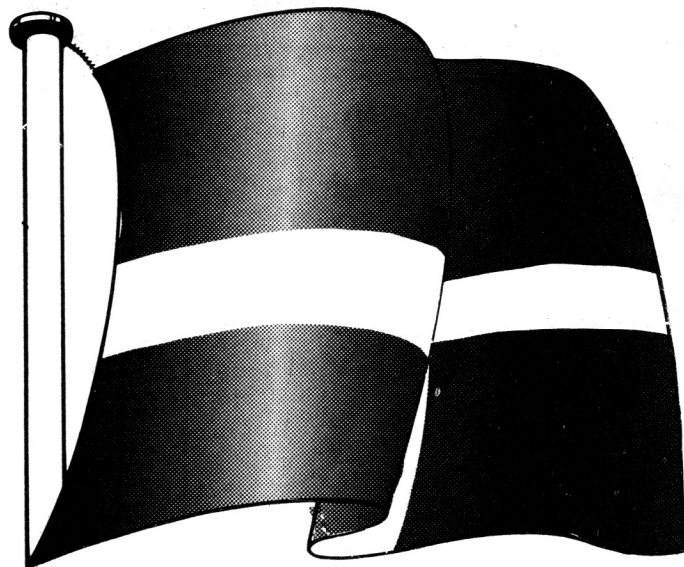
volde problemer for folk med bånd, for Bsic'en starter nu i adresse 820 hvor båndbufferen normalt har til huse. Men lad os se bort fra båndfolkene, og selv få det meste ud af det! Hvordan du starter det? jo, du skriver bare: sys50688:new (efterfulgt af RETURN!)

```
10 clr:restore:adr=50688
20 fort=1to2:chk=0:fore=1to80
30 reada:ifa=-1then50
40 chk=chk+a:pokeadr,a:adr=adr+1:nexste
50 reada:ifa<>chkthenprint"fejl i linjerne "50+(t*50)"-"90+(t*50):end
60 nextt
70 printchr$(147)chr$(18)" danske tegn ":print:print"lavet af crep i 1987"
80 print:print"aktiveres med: sys 50688":end
100 data 120,169,208,133,252,169,0,133,251,141,52,3,168,165,1,72
110 data 169,51,133,1,169,192,141,136,2,177,251,145,251,136,208,249
120 data 230,252,165,252,201,224,208,241,160,23,185,96,198,153,216,216
130 data 73,255,153,216,220,185,120,198,153,216,208,153,216,218,73,255
140 data 153,216,212,153,216,222,136,16,225,104,133,1,169,53,133,43,12565
150 data 169,3,133,44,169,148,141,0,221,169,5,141,24,208,88,96
160 data 0,0,118,27,127,216,126,0,0,0,59,110,126,118,220,0
170 data 24,0,60,6,62,102,62,0,31,60,108,127,108,108,111,0
180 data 59,110,110,126,118,118,220,0,24,0,60,102,126,102,102,-1,5352
```



Det frembringer et dansk flag på hele skærmen -også i borderen! Et skægt lille program til dine egne spil, introer

```
100 forx=49152to49399:reada:pokex,a:c=c+a:next
110 ifc<>32461thenprint"data fejl":end
120 printchr$(147)chr$(17)chr$(17)
130 print"sys 49152 for at starte dannebrog."
140 printchr$(17)chr$(17)
150 print"sys 49353 for at slukke dannebrog.":end
160 data 169,127,141,21,208,141,23,208,141,29,208,141,27,208,162,0,169,13,157
170 data 248,7,232,224,7,208,246,234,162,0,169,100,157,0,208,232,232,224,14
180 data 208,245,234,162,0,169,1,157,39,208,232,224,7,208,246,234,162,0,169
190 data 255,157,64,3,232,224,65,208,246,234,169,7,141,1,208,169,49,141,3,208
200 data 169,91,141,5,208,169,133,141,7,208,169,175,141,9,208,169,217,141,11
210 data 208,169,230,141,13,208,120,162,0,142,14,220,142,255,63,134,2,232,142
220 data 26,208,169,249,141,18,208,169,127,45,17,208,141,17,208,169,144,141
230 data 20,3,169,192,141,21,3,88,96,252,252,166,2,208,10,169,19,141,17,208
240 data 173,18,208,208,251,169,27,141,17,208,169,1,141,25,208,173,18,208,201
250 data 139,208,249,169,1,141,32,208,141,33,208,173,18,208,201,179,208,249
260 data 169,2,141,32,208,141,33,208,76,49,234,120,169,1,141,14,220,169,240
270 data 141,26,208,169,27,141,17,208,169,49,141,20,3,169,234,141,21,3,169
280 data 1,141,14,220,169,0,141,21,208,169,14,141,32,208,169,6,141,33,208,96
```



Dannebrog



# Hot space duel

Hot Space Duel! Det er et

to-spiller spil, hvor det gælder om, at svække modspilleren så meget, at han tilintetgøres. Samtidigt, skal man undgå at blive ramt af selvsammes sløvhedsbringende skydepiller. Det er svært og nervepirrende, men absolut et spil der er værd at taste ind!

```

10 clr:restore:adr=8192
20 fort=1to16:chk=0:fore=1to85
30 reada:ifa=-1then50
40 chk=chk+a:pokeadr,a:adr=adr+1:nexste
50 reada:ifa<>chkthenprint"fejl i linjerne "50+(t*50) "- "90+(t*50):end
60 nextt:sys8980
100 data 0,0,0,5,95,240,1,95,192,0,95,0,85,95,255,85,95
110 data 255,21,95,252,21,95,252,21,95,252,21,95,252,5,95,240,5
120 data 95,240,5,107,240,5,170,240,4,170,48,4,107,48,4,28,48
130 data 4,0,48,4,0,48,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
140 data 0,0,4,0,48,4,0,48,4,28,48,4,107,48,4,170,48,5642
150 data 5,170,240,5,107,240,5,95,240,5,95,240,21,95,252,21,95
160 data 252,21,95,252,21,95,252,85,95,255,85,95,255,0,95,0,1
170 data 95,192,5,95,240,0,0,0,0,169,70,141,4,208,169,130,141
180 data 6,208,169,200,141,5,208,169,70,141,7,208,169,13,141,248,7
190 data 141,249,7,169,129,141,250,7,169,128,141,251,7,169,15,141,28,9796
200 data 208,141,38,208,141,24,212,141,1,212,169,11,141,39,208,141,40
210 data 208,169,0,141,32,208,141,33,208,32,68,229,169,10,141,5,212
220 data 141,6,212,162,0,189,119,34,157,10,4,157,202,7,232,224,16
230 data 208,242,169,14,141,41,208,141,42,208,162,0,189,135,34,157,0
240 data 216,232,224,40,208,245,169,50,141,209,7,169,0,133,139,133,140,10499
250 data 169,12,141,28,208,141,21,208,141,37,208,162,0,138,157,64,3
260 data 232,224,64,208,248,169,60,141,91,3,141,93,3,141,94,3,141
270 data 96,3,173,0,220,201,123,208,3,206,4,208,201,119,208,3,238
280 data 4,208,201,111,208,29,165,139,201,1,240,23,32,44,34,238,21
290 data 208,169,190,141,1,208,173,4,208,234,234,141,0,208,169,1,133,10429
300 data 139,165,139,201,1,208,20,206,1,208,206,1,208,173,1,208,201
310 data 67,176,7,169,0,133,139,206,21,208,173,4,208,201,32,176,5
320 data 169,32,141,4,208,173,4,208,201,254,144,5,169,254,141,4,208
330 data 173,1,220,201,251,208,3,206,6,208,201,247,208,3,238,6,208
340 data 201,239,208,32,165,140,201,1,240,26,32,44,34,238,21,208,238,11386
350 data 21,208,169,90,141,3,208,173,6,208,234,234,141,2,208,169,1
360 data 133,140,165,140,201,1,208,23,238,3,208,238,3,208,173,3,208
370 data 201,204,144,10,169,0,133,140,206,21,208,206,21,208,173,6,208
380 data 201,32,176,5,169,32,141,6,208,173,6,208,201,254,144,5,169
390 data 254,141,6,208,173,30,208,201,9,208,6,206,25,4,238,42,208,11064
400 data 201,6,208,6,206,217,7,206,41,208,32,55,34,173,217,7,201

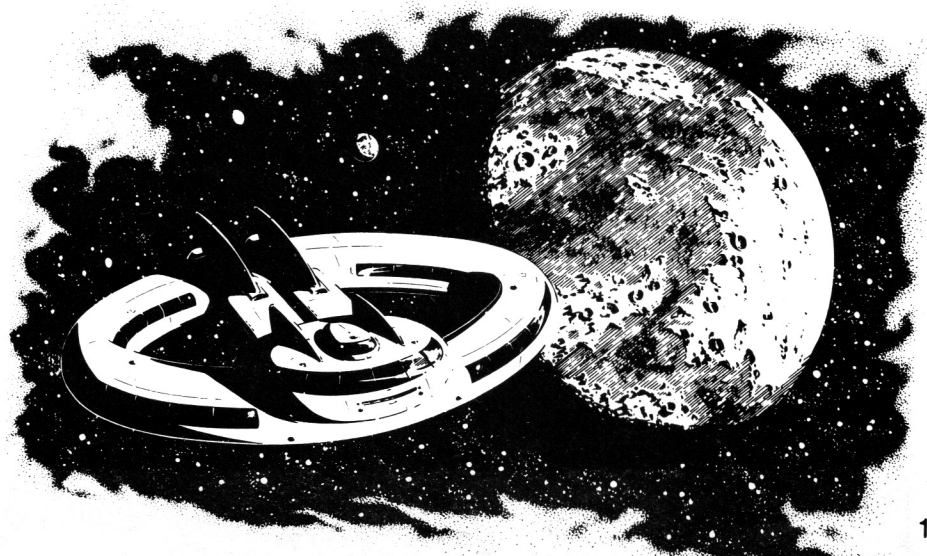
```

410 data 49,176,8,169,49,141,199,34,76,218,34,173,25,4,201,49,176  
420 data 8,169,50,141,199,34,76,218,34,76,35,33,169,0,141,4,212  
430 data 169,129,141,4,212,96,165,255,201,5,208,45,169,0,133,255,162  
440 data 40,189,0,216,157,1,216,202,224,40,208,245,162,0,189,193,219,10255  
450 data 157,192,219,232,224,40,208,245,173,38,216,141,2,216,141,230,219  
460 data 162,0,232,208,253,96,162,0,232,208,253,162,0,232,208,253,230  
470 data 255,96,16,12,1,25,5,18,32,49,32,16,15,23,5,18,58  
480 data 57,0,0,2,2,2,4,4,4,10,10,10,1,1,1,10,10  
490 data 10,4,4,4,2,2,2,0,0,0,0,0,0,0,0,0,0,0,6616  
500 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0  
510 data 0,0,0,0,0,0,0,16,12,1,25,5,18,32,49,32,23  
520 data 9,14,19,7,1,13,5,32,15,22,5,18,32,32,32,169  
530 data 5,141,134,2,32,68,229,162,0,189,192,34,157,195,5,189,205  
540 data 34,157,77,5,232,224,13,208,239,169,0,141,21,208,133,255,230,4955  
550 data 255,165,255,201,16,240,13,162,0,160,0,200,208,253,232,208,248  
560 data 76,252,34,234,234,234,120,169,27,141,17,208,169,1,141,26,208  
570 data 169,127,141,13,220,169,63,141,20,3,169,35,141,21,3,169,0  
580 data 133,2,133,255,169,40,160,36,32,30,171,88,76,248,36,173,25  
590 data 208,141,25,208,230,2,165,2,201,1,240,11,201,2,240,10,169,10574  
600 data 0,133,2,76,69,35,76,94,35,76,185,35,230,255,165,255,201  
610 data 63,208,4,169,0,133,255,166,255,24,169,20,125,122,35,141,18  
620 data 208,133,254,76,49,234,0,0,0,2,4,6,10,12,16,22,26  
630 data 32,36,42,48,54,60,64,70,76,82,88,92,98,102,106,110,112  
640 data 114,116,118,118,118,118,118,116,114,110,108,104,100,94,90,84,80,7973  
650 data 74,68,62,56,50,44,38,32,28,22,18,14,10,6,4,2,0  
660 data 0,164,254,200,200,162,0,189,219,35,204,18,208,208,251,141,32  
670 data 208,141,33,208,200,200,232,224,77,208,235,136,140,18,208,76,126  
680 data 234,9,0,0,9,9,0,9,9,9,9,8,9,9,8,8,9  
690 data 8,8,8,8,7,8,8,7,7,8,7,7,7,7,1,7,7,6151  
700 data 1,1,7,1,1,1,1,7,1,1,7,7,1,7,7,7,7  
710 data 8,7,7,8,8,7,8,8,8,8,9,8,8,9,9,8,9  
720 data 9,9,72,64,73,73,64,64,73,64,147,144,13,13,32,32,32  
730 data 32,32,32,32,32,32,32,32,62,32,72,79,84,32,83,80,65  
740 data 67,69,32,68,85,69,76,32,60,13,13,13,13,32,32,40,67,2806  
750 data 41,67,79,80,89,82,73,71,72,84,32,49,57,56,55,32,66  
760 data 89,32,65,76,76,65,78,32,80,69,84,69,82,83,69,78,13  
770 data 13,13,13,13,32,32,32,32,32,32,32,32,32,32,32,32,32  
780 data 32,32,67,79,78,84,82,79,76,83,58,13,13,13,32,32,32  
790 data 32,32,32,32,32,80,76,65,89,69,82,32,49,45,74,79,89,4567  
800 data 83,84,73,67,75,32,80,79,82,84,32,49,13,13,32,32,32  
810 data 32,32,32,32,32,80,76,65,89,69,82,32,50,45,74,79,89  
820 data 83,84,73,67,75,32,80,79,82,84,32,50,13,13,13,13,13  
830 data 32,32,32,32,32,32,32,32,32,32,72,73,84,32,83,80,65  
840 data 67,69,32,84,79,32,71,79,32,79,78,33,33,0,165,197,201,4958  
850 data 60,208,250,120,169,240,141,26,208,169,49,141,20,3,169,234,141  
860 data 21,3,169,0,141,13,220,88,76,128,32,-1,3239

**Program: Hot Space Duel**

**Gevinst: Indoor Sports  
Boulder Dash Constr. Kit  
BWW-tekst 1**

**Udstyr: C-64/128**





Endnu et spil... Denne gang, er det nu en klassiker af den art, vi kender fra BR-butikkerne. Nemlig et 4 på stribe-spil. Hvordan det spilles, giver såmænd sig selv, men der er 3 forskellige modes: Spiller mod spiller, spiller mod computer, eller computer mod computer. Den sidstnævnte er absolut værd at bemærke - Det er meget morsomt at se computeren udspille sig selv!!

Programmet er leveret af:

Anders Jensen  
Gl. Skørpingevej 68  
9520 Skørping

**Program: 4 På Stribe**  
**Gevinst: Indoor Sports**  
**Boulder Dash Constr. Kit**  
**Udstyr: C-64/128**

```

5 clr:gosub 4000:gosub 2000
10 poke 53280,0:poke 53281,0:printchr$(28)chr$(147)chr$(142)chr$(8)
20 l=54272:for i=1 to l+23:poke i,0:next:poke i,5:poke l+6,245
30 i=rnd(-ti):def fn t(x)=int(x*rnd(1))
40 tf=1:ra=1440:dim f(6),s1(6)
50 s$="spiller ":d$="datamat ":e$="1":t$="2":data 1,41,40,39,0
60 mn$=chr$(19):fort=1to19:mn$=mn$+chr$(17):next
100 print chr$(147)chr$(17)ic$(0)chr$(156)chr$(18)"angiv venligst:"
110 print ic$(1)chr$(17)chr$(18)"1"chr$(146)":chr$(158)s$mod "s$
120 print ic$(1)chr$(18)"2"chr$(146)":chr$(158)s$mod "d$
130 print ic$(1)chr$(18)"3"chr$(146)":chr$(158)d$mod "d$:j$="123"
140 print ic$(1)right$(ic$(0),3)chr$(129)"spiltype ";:gosub 1000:ty=val(i$)
150 ifp(1)orp(2)thenprintic$(2);:j$="jn":gosub1000:ifi$="j"thenps(1)=0:ps(2)=0
160 ifty=1 then s$(1)=s$+e$:s$(2)=s$+t$
170 ifty=2 then s$(1)=s$:s$(2)=d$
180 ifty=3 then s$(1)=d$+e$:s$(2)=d$+t$
190 s$(1)=chr$(31)+s$(1):s$(2)=chr$(5)+s$(2)
200 print chr$(147)tab(14)chr$(5)chr$(18)"4 paa stribe"
220 print ic$(3)tab(16)chr$(153)"abcdefg"chr$(159)chr$(17)
230 for i=1 to 7:print tab(15)"B      B":next
240 print tab(15)"JCCCCCCK"
250 f=1+5*fn t(2):s=2+(f=6):je=0:j$="abcdefg"
300 if je=49 then m$="det blev uagjort":goto 540
310 f=7-f:s=3-s:p$=left$(mn$,4+s+s)+s$(s)+"      "+ic$(4):print p$;
320 on 2+(ty=1)+(ty=2 and s=1) gosub 600,700:je=je+1
400 restore:vf=f

```



```

410 read r:if r=0 then 300
420 gosub 760:if jt<99 then 410
500 m$=s$(s)+" vandt":ps(s)=ps(s)+1
510 if peek(p+r)=81 and (peek(p+r+tf) and 15)=f then p=p+r:goto 510
520 poke l+4,33:for i=1 to 9:forj=0to3:pokel+1,40+3*j:pokep-r*j,290-peek(p-r*j)
530 for k=0 to 9:nextk,j,i:poke l+4,32
540 print mn$tab(19-int(len(m$)/2))chr$(18)chr$(158)m$
545 if ty=1 then print ic$(5)"spiller 1:"ps(1)"    spiller 2:"ps(2)
550 if ty=2 then print ic$(5)"spiller : "ps(1)"    datamat : "ps(2)
555 if ty=3 then print ic$(5)"datamat 1:"ps(1)"    datamat 2:"ps(2)
560 print chr$(17)chr$(156)"vil du spille igen    "ic$(6);
570 j$="jno":gosub1000:if i$="j" then 200
580 if i$="o" then ps(1)=0 and ps(2)=0: goto 100
590 print chr$(147):end
600 gosub 1000:js=asc(i$)-65:if peek(ra+40+js)=32 then 900
610 poke l+1,20:poke l+4,33:for i=0 to 99:next:poke l+4,32:for i=0 to 200:next
620 print p$;:goto 600
700 print " : ";b=-1:for i=0 to 6:restore:s0=0:p=ra+40+i:if peek(p)=81 then 750
710 if peek(p+40)=32 then p=p+40:goto 710
720 read r:if r then vf=f:gosub 760:s0=s0+jt:vf=24-f:gosub 760:s0=s0+jt:goto720
730 if s0>b then b=s0:ls=0
740 if s0=b then sl(ls)=i:ls=ls+1
750 next:js=sl(fn t(ls)):print chr$(js+65):goto 900
760 jt=0
770 q=p:for j=1 to 2
780 if peek(p+r)=81 and (peek(p+r+tf) and 15)=vf then jt=jt+1:p=p+r:goto 780
790 if jt>3 then jt=3
800 p=q:r=-r:next:jt=jt-99*(jt=3):return
900 p=ra+js:lt=35:poke l+4,17
910 poke l+1,lt:poke p+tf,f:poke p,81:for i=0 to 99:next
920 if peek(p+40)=32 then poke p,32:p=p+40:lt=lt-1:goto 910
930 poke l+4,16:return
1000 print " : ";:poke 204,0:poke 198,0
1010 get i$:for i=1 to len(j$):if i$<>mid$(j$,i,1) then next:goto 1010
1020 poke 204,1:print i$:return
2000 print chr$(147):poke 53280,14:poke53281,6
2005 gosub 2080
2010 for t=1 to 4
2020 if t=1 then 2040
2025 if t=3 then 2045
2030 if t=2 then 2050
2035 if t=4 then 2055
2040 print ic$(7)chr$(5)ic$(0)"4 paa stribe!"
2045 print ic$(7)chr$(155)ic$(0)"4 paa stribe!"
2050 print ic$(7)chr$(144)ic$(0)"4 paa stribe!"
2055 print ic$(7)chr$(144)ic$(0)"4 paa stribe!"
2060 next t
2070 gete$:ife$=""then2010
2073 if e$=" " thenreturn
2080 print ic$(3)ic$(0)right$(ic$(0),2)chr$(5)"lavet af "
2090 print ic$(0)"anders jensen"
3000 print ic$(0)"november 1987"
3010 return
4000 ic$(0)=chr$(17)+chr$(17):fort=1to13:ic$(0)=ic$(0)+chr$(29):next
4010 ic$(1)=chr$(30)+right$(ic$(0),10):ic$(2)=chr$(155)+chr$(17)+"points = 0"
4020 fort=1to6:ic$(3)=ic$(3)+chr$(17):next:ic$(4)=chr$(157)+chr$(157)+chr$(157)
4030 ic$(5)=chr$(129)+chr$(17):ic$(6)=chr$(18)+"o"+chr$(146)+"m/"+chr$(18)
4040 ic$(6)=ic$(6)+"j"+chr$(146)+"a/"+chr$(18)+"n"+chr$(146)+"ej:"+chr$(144)
4050 ic$(7)=chr$(19)+left$(ic$(3),3)
4060 return

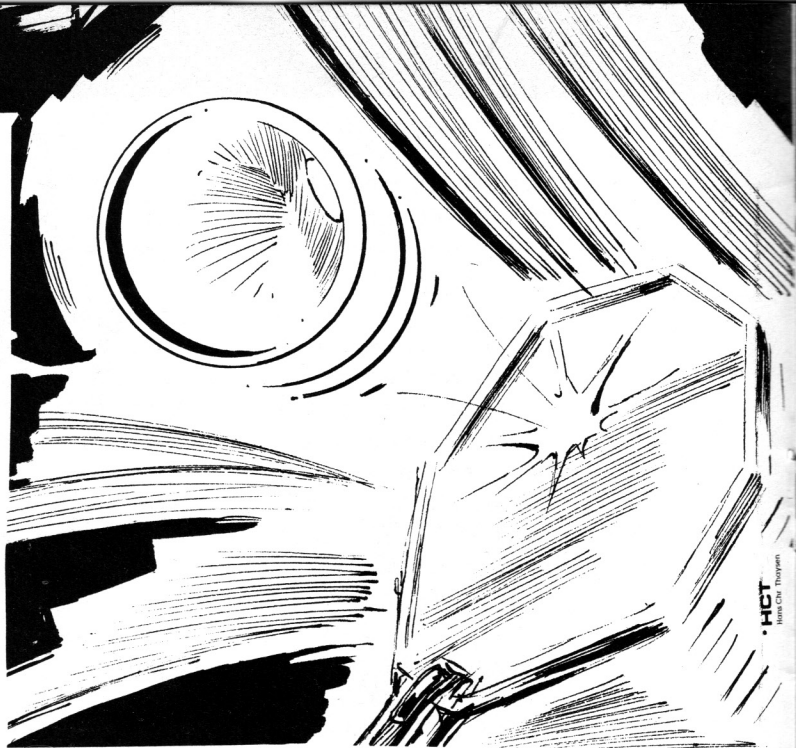
```

## Sprite Bouncer

En rutine egnet til intro'er, startskærme, spil osv. Hvad kan den? - Den kan få en sprite til at bevæge sig rundt på skærmen på de mest fantastiske måder! Den bevæger sig endda helt glat! Man kan få en sprite til at køre ud til siden for så at skifte retning, og kører ud til den anden side - hastigheden bestemmer du selv! Du skal selv definere spriten, sætter farverne, pointerne osv., for rutinen ændrer kun x- og ypositionerne, dvs. adresserne \$d000-\$d001 (#53248-#53249). De følgende registre bestemmer retningen og hastigheden:

```
#251 ($fb) = op
#252 ($fc) = ned
#253 ($fd) = venstre
#254 ($fd) = højre
```

```
10 clr:restore:adr=49700
20 fort=1to4:chk=0:fore=1to80
30 reada:ifa=-1then50
40 chk=chk+a:pokeadr,a:adr=adr+1:nexthe
50 reada:ifa<>chkthenprint"fejl i linjerne "50+(t*50)"-"90+(t*50):end
60 nextt
70 printchr$(147)chr$(18)" sprite bouncer ":print:print"lavet af crep i 1987"
80 print:print"aktiveres med: sys 49703"
90 print:print"deaktiveres med: sys 49700":end
100 data 76,253,194,120,169,72,141,20,3,169,194,141,21,3,169,1
110 data 141,26,208,169,127,141,13,220,169,27,141,17,208,169,255,141
120 data 18,208,88,96,169,1,141,25,208,169,255,141,18,208,165,251
130 data 240,10,165,252,240,6,169,0,133,251,133,252,165,253,240,10
140 data 165,254,240,6,169,0,133,253,133,254,173,1,208,56,229,251,11123
150 data 141,1,208,173,1,208,24,101,252,141,1,208,173,0,208,56
160 data 229,253,141,0,208,176,8,173,16,208,73,1,141,16,208,173
170 data 0,208,24,101,254,141,0,208,144,8,173,16,208,73,1,141
180 data 16,208,173,1,208,201,230,144,15,169,229,141,1,208,165,252
190 data 133,251,169,0,133,252,240,17,201,50,176,13,169,50,141,1,9977
200 data 208,165,251,133,252,169,0,133,251,174,0,208,173,16,208,74
210 data 144,19,224,65,144,32,169,64,141,0,208,165,254,133,253,169
220 data 0,133,254,240,17,224,24,176,13,169,24,141,0,208,165,253
230 data 133,254,169,0,133,253,76,49,234,120,169,49,141,20,3,169
240 data 234,141,21,3,169,129,141,13,220,169,240,141,26,208,169,27,10663
250 data 141,17,208,88,96,-1,550
```



Hvis vi f.eks. vil have sprit'en til at rykke opad for derefter at rykke nedad, skal vi sætte op-pointeren til den hastighed, vi vil have den til at køre. Hvis vi vil have den til at rykke fem linjer op, vil vi sætte pointeren til 5. Samtidig vil vi gerne have sprit'en til at rykke sig med en hastighed på to til venstre, så venstre-pointeren sætter vi til 2. Det gør vi således: poke251,5:poke252,0:poke253,2:poke254,0 (efterfulgt af RETURN). Du kan selv prøve at eksperimentere - det bliver under alle omstændigheder seje effekts du får ud af det!

# LOCK KEYS

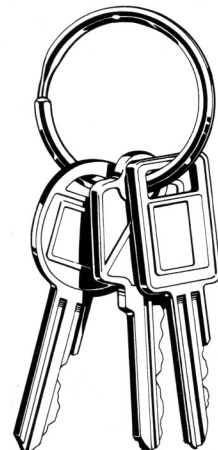


## Lock Keys

Har man lavet et program, hvor man bruger input, get eller lignende tingester, hvor man går ud fra inputet fra tastaturet, kan det til tider være irriterende, at man ikke kan afbryde enkelte taster, såsom cursor-knapperne, clr/home osv. Det kan man i realiteten godt, men det kræver et lille programtillæg. Det bringer vi her i form af et program, som gør det muligt at låse op til 7 (syv) forskellige knapper ad gangen. Du skal dog være opmærksom på, at du hver gang SKAL angive syv knapper der skal låses. Hvis du f.eks. kun vil låse to, må du fylde de resterende fem ud med de samme. Værdierne for de knapper du vil låse, er ikke ASCII værdier, men de værdier computeren læser fra tastaturet (dvs. der bliver ikke taget hensyn til SHIFT-, LOGO- eller CTRLknapperne.) De forskellige knappers talværdier er dem som figurerer i adressen 203 eller \$00cb i hex

program: Lock Keys

```
10 clr:restore:adr=820
20 fort=1to3:chk=0:fore=1to85
30 reada:ifa=-1then50
40 chk=chk+a:pokeadr,a:adr=adr+1:nexthe
50 reada:ifa<>chkthenprint"fejl i linjerne "50+(t*50)"-"90+(t*50):end
60 nextt
70 printchr$(147)chr$(18)" lock keys ":print:print"lavet af crep i 1987"
80 print:print"aktiveres med: sys 820,1,a,b,c,d,e,f,g"
90 print:print"deaktiveres med: sys 820,0":end
100 data 32,241,183,224,2,144,6,32,69,3,76,72,178,224,1,240,18
110 data 169,72,141,143,2,169,235,141,144,2,96,0,0,0,0,0
120 data 0,169,210,141,143,2,169,3,141,144,2,32,241,183,224,64,144
130 data 6,32,69,3,76,72,178,142,80,3,32,241,183,224,64,144,6
140 data 32,69,3,76,72,178,142,81,3,32,241,183,224,64,144,6,32,8208
150 data 69,3,76,72,178,142,82,3,32,241,183,224,64,144,6,32,69
160 data 3,76,72,178,142,83,3,32,241,183,224,64,144,6,32,69,3
170 data 76,72,178,142,84,3,32,241,183,224,64,144,6,32,69,3,76
180 data 72,178,142,85,3,32,241,183,224,64,144,6,32,69,3,76,72
190 data 178,142,86,3,96,165,203,205,80,3,240,30,205,81,3,240,25,8415
200 data 205,82,3,240,20,205,83,3,240,15,205,84,3,240,10,205,85
210 data 3,240,5,205,86,3,208,4,169,64,133,203,76,72,235,-1,3634
```



# EDITOR

## Editor

Dette program vil uden tvivl gøre din hverdag nemmere. Men (ja, der er et men), kun hvis du har noget at bruge det til! Er du programmør - eller bare freak - med flair for det grafiske, så kan du godt gribe det nærmeste joystick og smide det i computeren! Det skal du nemlig bruge, når du skal til at designe dine egne spil baner med din egen grafik! Grafikken kan du ikke tegne hermeeen, du kan designe banen! Denne editor frigør et areal på 256 x 96 som du kan bruge til at tegne en eller flere baner på. Du loader dit karaktersæt ind i adresse \$3000 (#12288), og banen ligger fra \$4000 (#16384) til \$9fff (#40959). Ved at bruge joystick i port 2, kan du vælge hvilke karakterer du vil bruge, og ved at trykke på fiknappen, kommer de ud på banen. Du kan også taste karaktererne ind på banen, ligesom når du normalt skriver på skærmen. Editoren har selvfølgelig også nogle funktioner indbygget på forskellige knapper:

f1 : ændrer karakterfarven  
f3 : ændrer multicolor 1  
f5 : ændrer multicolor 2  
f7 : ændrer baggrundsfarven  
f2 : ændrer farven på banecursoren  
f4 : ændrer farven på karaktercursoren  
RUN/STOP : stopper editoren  
CLR/HOME : får cursoren til at stå på første linje  
SHIFT + CLR/HOME : sletter hele banen  
SHIFT + INST/DEL : sletter den linje cursoren står på

I øvrigt bruges cursor-knapperne til at flytte rundt på banen, ligesom ved normal brug. Editoren kan også bruges som en slags tekstbehandling, hvis man sætter karakterfarven til ikke at være i multicolor. Dog kan man ikke printe teksten ud, men man kan jo selv lave en lille rutine der gør dette.

program: Editor

10 clr:restore:adr=49152

20 fort=1to10:chk=0:fore=1to80

30 reada:ifa=-1then50

40 chk=chk+a:pokeadr,a:adr=adr+1:nexthe

50 reada:ifa<>chkthenprint"fejl i linjerne "50+(t\*50)"-"90+(t\*50):end

60 nextt

70 printchr\$(147)chr\$(18)" editor ":print:print"lavet af crep i 1987"

80 print:print"aktiveres med: sys 49152"

90 print:print"deaktiveres med tryk paa run/stop":end

100 data 169,13,141,248,7,141,249,7,169,112,141,1,208,169,53,141

120 data 2,208,169,176,141,3,208,169,181,141,0,208,169,3,141,21

130 data 208,32,68,229,169,0,141,16,208,32,212,194,173,33,208,141

140 data 32,208,169,28,141,24,208,169,216,141,22,208,160,62,185,133

150 data 194,153,64,3,136,16,247,169,132,133,251,141,209,194,169,6,10296

160 data 200,192,32,208,246,72,165,251,24,105,40,133,251,144,2,230

170 data 252,104,202,16,228,173,4,194,141,162,192,174,3,194,142,161

180 data 192,169,4,141,165,192,169,0,141,164,192,169,16,133,2,160

190 data 88,173,162,192,201,64,48,4,201,160,48,4,169,32,208,3,10357

200 data 173,0,80,141,0,4,238,161,192,238,162,192,198,2,208,209,206,211

210 data 192,200,16,221,142,161,192,238,162,192,194,173,0,220,74,176

220 data 194,48,3,76,117,193,169,3,141,211,194,173,0,220,74,176

230 data 29,72,173,3,208,201,176,240,20,56,233,8,141,3,208,173

240 data 209,194,56,233,40,141,209,194,176,3,206,210,194,104,74,176

250 data 29,72,173,3,208,201,232,240,20,24,105,8,141,3,208,173

260 data 209,194,24,105,40,141,209,194,144,3,238,210,194,104,74,176

270 data 30,72,173,2,208,201,53,240,21,56,233,8,141,2,208,176

280 data 5,169,0,141,16,208,206,209,194,208,3,206,210,194,104,74

290 data 176,30,72,173,2,208,201,45,240,21,24,105,8,141,2,208,9726

300 data 144,5,169,2,141,16,208,238,209,194,133,252,160,0,177,251,141

310 data 72,174,209,194,134,251,173,210,194,133,252,160,0,153,0

320 data 170,6,104,74,176,15,173,170,6,141,168,6,32,9,194,32

330 data 162,0,32,19,194,173,1,220,201,127,208,3,76,196,194,32

340 data 159,255,24,32,35,194,32,228,255,240,57,201,147,240,50,201,10776

350 data 148,240,46,32,210,255,160,0,162,1,201,145,208,3,32,27

360 data 194,201,17,208,3,32,19,194,202,201,157,208,3,32,27,194

370 data 201,29,208,9,140,0,80,142,1,80,32,19,194,192,0,208

380 data 3,32,42,194,169,0,201,5,208,25,169,32,160,0,153,0

390 data 64,200,208,250,238,208,193,174,208,193,224,160,208,240,169,64,9691

400 data 141,208,193,162,1,173,4,194,201,56,176,6,32,19,194,76

410 data 249,193,201,152,144,3,32,27,194,192,127,208,3,76,190,192

420 data 76,117,192,0,64,0,0,64,160,0,173,168,6,141,20

430 data 194,200,96,160,0,162,17,76,240,255,201,148,208,18,160,0,9569

440 data 169,32,32,14,194,136,162,0,32,19,194,192,0,208,243,96

450 data 201,19,208,21,169,0,141,3,194,169,64,141,4,194,169,20

460 data 141,15,194,169,72,141,16,194,96,201,147,208,28,160,0,169

470 data 32,162,64,142,104,194,153,0,96,200,208,250,238,104,194,174

480 data 104,194,224,160,208,240,76,68,194,76,229,194,32,9,194,162,10070

490 data 0,32,19,194,96,255,252,0,255,252,0,224,28,0,224,28,0

500 data 0,224,28,0,224,28,0,224,28,0,0,0,0,0,0,0

510 data 224,28,0,255,252,0,255,252,0,0,0,0,0,0,0

520 data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

530 data 0,0,0,0,120,169,49,141,20,3,169,234,141,21,3,88,5543

540 data 0,0,0,0,3,169,196,141,136,2,173,134,2,32,68,229,169

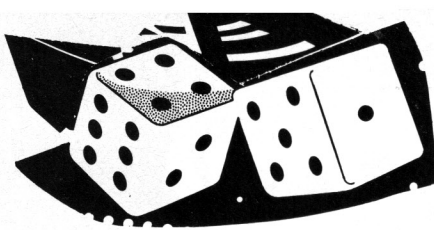
550 data 96,0,0,3,169,196,201,133,208,6,238,134,2,76,212,194,201

560 data 4,141,136,2,96,201,133,208,4,238,35,208,96,201

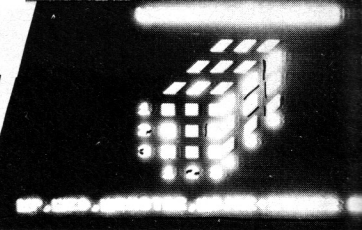
570 data 134,208,4,238,34,208,96,201,135,208,4,238,35,208,96,201

580 data 136,208,7,238,32,208,238,33,208,96,201,137,208,4,238,39

590 data 208,96,201,138,240,3,76,124,194,238,40,208,96,-1,9875



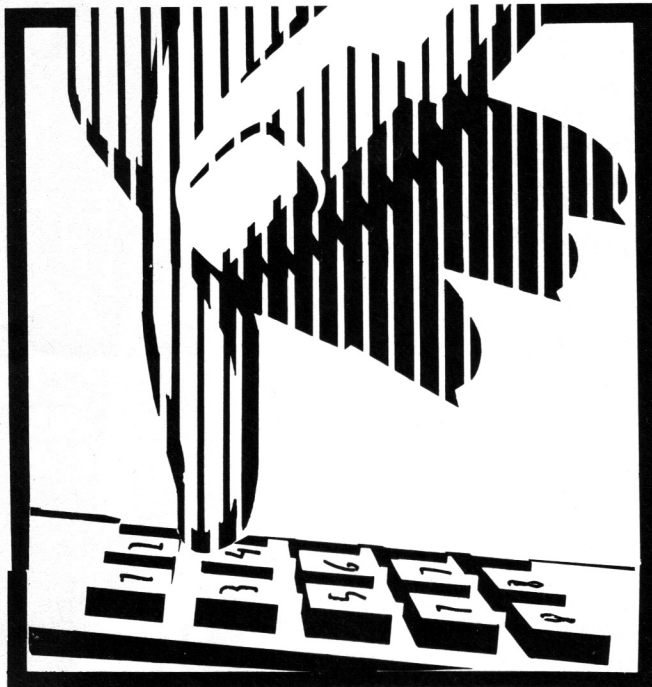
# TERNING



**Program:** Professor Terning  
**Gevinst:** Indoor Sports  
**Boulder Dash Constr. Kit**  
**Udstyr:** C-64/128

Programmøren er:  
 Rene Nielsen  
 Egevej 21  
 4970 Rødby

Vi kender velsagtens allesammen den dersens Professor Terning, -ikke? Her har vi den nu, i en computerversion. Den vil sikkert vække minderne hos de, der for år tilbage, og måske endnu, kunne sidde og bruge timer og dage på at prøve at få den s... terning til at blive normal igen! Programmet er skrevet i BASIC, men det er alligevel hurtigt og velfungerende!



# F-TOOLS

**Program:** F-Tools  
**Gevinst:** Boulder Dash Constr. Kit  
**BWW-tekst 1**  
**Udstyr:** C-64/128

Med denne lille, fikse rutine, kan du bruge f-tasterne til at få udskrevet nogle af de hyppigst benyttede kommandoer i 64'eren's BASIC. De forskellige kommandoer er placeret på de følgende taster:

**f1 = list    f2 = load**  
**f3 = for    f4 = save**  
**f5 = next    f6 = then**  
**f7 = (run + f8 = goto**  
**RETURN)**

Du trykker bare på vedkommende tast, og Vupti... Den pågældende kommando bliver lynhurtigt skrevet ud på skærmen!!

Rutinen er indsendt af:  
 Anders Buch  
 Rolighedsvej 28  
 7790 Hvidbjerg

```

10 clr:restore:adr=49152
20 fort=1to4:chk=0:fore=1to85
30 reada:ifa=-1then50
40 chk=chk+a:pokeadr,a:adr=adr+1:nexte
50 reada:ifa<>chkthenprint"fejl i linjerne "50+(t*50)"-"90+(t*50):end
60 nextt:printchr$(147)chr$(18)" f-tools ":print:print"lavet af ab-software"
70 print:print"startes med: sys 49152":print:print"slukkes med: sys 49170":end
100 data 120,169,32,141,20,3,169,192,141,21,3,88,96,0,0,0,0
110 data 0,120,169,49,141,20,3,169,234,141,21,3,88,96,0,165,197
120 data 197,2,208,3,76,49,234,133,2,201,4,240,24,201,5,240,11
130 data 201,6,240,10,201,3,240,9,76,49,234,76,142,192,76,203,192
140 data 76,8,193,169,1,205,141,2,240,32,169,76,141,119,2,169,73,8607
150 data 141,120,2,169,83,141,121,2,169,84,141,122,2,169,32,141,123
160 data 2,169,5,133,198,76,49,234,169,76,141,119,2,169,79,141,120
170 data 2,169,65,141,121,2,169,68,141,122,2,169,32,141,123,2,169
180 data 5,133,198,76,49,234,169,1,205,141,2,240,22,169,70,141,119
190 data 2,169,79,141,120,2,169,82,141,121,2,169,3,133,198,76,49,8912
200 data 234,169,83,141,119,2,169,65,141,120,2,169,86,141,121,2,169
210 data 69,141,122,2,169,32,141,123,2,169,5,133,198,76,49,234,169
220 data 1,205,141,2,240,27,169,78,141,119,2,169,69,141,120,2,169
230 data 88,141,121,2,169,84,141,122,2,169,4,133,198,76,49,234,169
240 data 84,141,119,2,169,72,141,120,2,169,69,141,121,2,169,78,141,9204
250 data 122,2,169,4,133,198,76,49,234,169,1,205,141,2,240,27,169
260 data 82,141,119,2,169,85,141,120,2,169,78,141,121,2,169,13,141
270 data 122,2,169,4,133,198,76,49,234,169,71,141,119,2,169,79,141
280 data 120,2,169,84,141,121,2,169,79,141,122,2,169,4,133,198,76
290 data 49,234,-1,7529

```



```

0 rem**professor terning**
2 a1=1:a2=2:a3=3:printchr$(147)chr$(5):poke53281,12:dimt(3,3,6):d=1356
4 e=55586:forx=1to6:readx3:forx1=1to3:forx2=1to3:t(x2,x1,x)=x3
6 nextx2,x1,x:g=55361:h=55639:gosub110
10 fory=40to200step80:forx=0to2:poked+x*2+y,160:nextx,y
12 fory=42to198step78:forx=2to6step2:poked+x-y,233:poked+x+1-y,105:nextx,y
14 fory=1to3:forx=2to6step2:poked+3*x*40-y*78,233:poked+43+x*40-y*78,105
16 nextx,y:printic$(1)ic$(10)"professor-terning"ic$(1)
18 ifv=0then30
20 printic$(0)"3"ic$(3)"2"ic$(3)"1"ic$(4)"3"ic$(5)"2"ic$(5)"1"ic$(6)"6";
21 printic$(7)"5"ic$(7)"4":a1=6:a2=5:a3=4:v=1
22 fory=1to3:forx=1to3:pokee+2*x+80*y,t(x,y,a1):nextx,y
24 fory=1to3:forx=2to6step2:pokeg-x+78*y,t(x/2,y,a2):pokeg-x-1+78*y,t(x/2,y,a2)
25 nextx,y
26 fory=1to3:forx=2to6step2:pokeh+40-x*40+y*78,t(x/2,y,a3)
27 pokeh-x*40+y*78,t(x/2,y,a3)
28 nextx,y:goto40
30 printic$(0)"1"ic$(3)"2"ic$(3)"3"ic$(4)"1"ic$(5)"2"ic$(5)"3"ic$(6)"4";
31 printic$(7)"5"ic$(7)"6":a1=1:a2=2:a3=3:v=0
32 fory=1to3:forx=1to3:pokee+2*x+80*y,t(x,y,a1):nextx,y
34 fory=1to3:forx=2to6step2:pokee+78+x-78*y,t(x/2,y,a2)
35 pokee+78+x+1-78*y,t(x/2,y,a2):nextx,y
36 fory=1to3:forx=2to6step2:pokee+45+x*40-y*78,t(x/2,y,a3)
37 pokee+85+x*40-y*78,t(x/2,y,a3)
38 nextx,y
40 printic$(8)ic$(10)"o"ic$(p,"ic$(10)"n"ic$(ed,"ic$(10)"v"ic$(enstre,";
41 printic$(10)"h"ic$(0jre+nummer el."ic$(10)"d"ic$(rej?"ic$(9)" ";
42 geta$:ifa$=""or(a$<"o"anda$<"v"anda$<"h"anda$<"d"anda$<"n")then 42
44 ifa$="d"andv=0then20
46 ifa$="d"andv=1then30
48 printa$;
50 getb$:ifb$=""then 50
52 ifasc(b$)=20thenprintb$;:goto42
54 ifasc(b$)<49orasc(b$)>54then50
56 b=val(b$):ifa$="o"andb<4theni=5:j=6:k=2:l=1:goto66
58 ifa$="v"theni=3:j=6:k=4:l=1:m=3:n=1:o=1:p=2:r=2:q=3:z=1:goto92
60 ifa$="h"theni=3:j=6:k=4:l=1:m=1:n=3:o=2:p=1:r=3:q=2:z=3:goto92
62 ifb>3thenb=b-3:goto76
64 i=1:j=2:k=6:l=5
66 forx=1to3:a4=t(b,x,i):t(b,x,i)=t(b,4-x,j):t(b,4-x,j)=t(b,4-x,k)
68 t(b,4-x,k)=t(b,x,l):t(b,x,l)=a4:next:ifb=2then18
70 c=3:ifb=1thenc=4
72 ifa$="o"thenm=3:n=1:o=1:p=2:r=2:q=3:goto86
74 m=1:n=3:o=2:p=1:r=3:q=2:goto86
76 ifa$="o"theni=3:j=5:k=4:l=2:m=3:n=1:o=1:p=2:r=2:q=3:goto80
78 i=2:j=4:k=5:l=3:m=1:n=3:o=2:p=1:r=3:q=2
80 forx=1to3:a4=t(x,b,i):t(x,b,i)=t(4-x,b,j):t(4-x,b,j)=t(4-x,b,k)
81 t(4-x,b,k)=t(x,b,l)
82 t(x,b,l)=a4:next:ifb=2then18
84 c=1:ifb=3thenc=6
86 a(1)=t(1,1,c):t(1,1,c)=t(m,n,c):t(m,n,c)=t(3,3,c):t(3,3,c)=t(n,m,c)
87 t(n,m,c)=a(1)
88 a(1)=t(o,p,c):t(o,p,c)=t(p,o,c):t(p,o,c)=t(q,r,c):t(q,r,c)=t(r,q,c)
89 t(r,q,c)=a(1)
90 goto18
92 fory=1toz:forx=1to3:a4=t(b,x,i):t(b,x,i)=t(4-x,b,j):t(4-x,b,j)=t(b,4-x,k)
94 t(b,4-x,k)=t(x,b,l):t(x,b,l)=a4:nextx,y:ifb=2then18
96 c=2:ifb=3thenc=5
98 goto86
100 data1,2,7,6,4,8
110 ic$(1)=chr$(19):fort=1to10:ic$(1)=ic$(1)+chr$(29):next:ic$(10)=chr$(18)
120 fort=1to9:ic$(2)=ic$(2)+chr$(17):next:ic$(0)=ic$(1)+ic$(2):ic$(5)=chr$(29)
125 ic$(3)=chr$(157)+chr$(17)+chr$(17):ic$(8)=chr$(19)+ic$(2):ic$(146)
130 ic$(4)=chr$(17)+chr$(17)+ic$(5):ic$(6)=chr$(145):ic$(9)=chr$(157)+chr$(157)
140 ic$(7)=ic$(5)+ic$(6)+ic$(6):fort=1to9:ic$(8)=ic$(8)+chr$(17):next
150 return

```



## Dec To Hex

Hvis man sidder og skal have regnet et tal fra decimal om til hexadecimal, hvad skal man så med en speciel regnemaskine, når din computer kan klare det for dig? Du kan regne enkelte tal og selvfølgelig

også komplicerede regnestykker om til hexadecimal. Prøv f.eks.:  
sys832,peek(44)\*256 + peek(43) for at få udskrevet basic-starten i hex. Et smart og nyttigt program for alle computerfreaks!

## Erazor

Lad ikke navnet skræmme dig - det har ikke noget med Motorsavsmassakren at gøre. Grunden til, at dette program hedder Erazor, er såmænd bare for at hentyde til et barberblad, som jo er bekendt er knivskarpt! Det er dette program også! Nemlig fordi det sletter skærmen, som var det en økse der flækker et hår i to! Du starter programmet, og det der står på skærmen, glider blødt og knivskarpt ind under den nederste border, hvorefter skærmen vender tilbage til normalt, men denne gang er den slettet! Det kan tilføre alle dine programmer nogle seje og ophidsende momenter!



```
program: Erazor
```

```
10 clr:restore:adr=49152
20 fort=1to2:chk=0:fore=1to75
30 reada:ifa=-1then50
40 chk=chk+a:pokeadr,a:adr=adr+1:nexste
50 reada:ifa<>chkthenprint"fejl i linjerne "50+(t*50)"-"90+(t*50):end
60 nextt
70 printchr$(147)chr$(18)" erazor ":print:print"lavet af crep i 1987"
80 print:print"aktiveres med: sys 49152":end
100 data 120,169,102,141,20,3,169,192,141,21,3,169,127,141,13
110 data 220,169,1,141,26,208,169,27,141,17,208,169,40,141,18
120 data 208,169,1,141,25,208,173,255,63,141,145,192,169,0,141
130 data 255,63,169,4,133,251,88,173,18,208,201,255,208,249,165
140 data 251,201,207,208,243,32,68,229,120,169,49,141,20,3,169,9807
150 data 234,141,21,3,169,240,141,26,208,169,129,141,13,220,169
160 data 27,141,17,208,173,145,192,141,255,63,88,96,120,166,251
170 data 240,23,234,234,234,173,18,208,41,7,9,24,141,17,208
180 data 160,7,136,208,253,234,234,202,208,233,169,40,141,18,208
190 data 169,1,141,25,208,230,251,76,126,234,-1,9830
```



```

10 adr=832
20 reada:ifa=-1then40
30 pokeadr+t,a:t=t+1:chk=chk+a:goto20
40 ifchk<>4330thenprint"der er en datafejl!":end
50 printchr$(147)chr$(18)" dec to hex ":print:print"lavet af crep i 1987"
60 print:print"aktiveres med: sys 832,tal":end
100 data 32,253,174,32,138,173,32,247,183,169,36,32,210,255,165,21,32
110 data 85,3,165,20,72,74,74,74,74,32,96,3,104,41,15,201,10
120 data 144,2,105,6,105,48,32,210,255,96,-1

```



### Bånd Salat

Ikke uden grund, har jeg kaldt dette program for »Bånd Salat«. Du kan nemlig, med dette program, høre musik fra din datasette. Du smider et musikbånd i, starter dette program, spænder hørebrillerne for, og du kan høre den musik der er på båndet! Kvaliteten på musikken afhænger meget hvor godt dit udstyr er. Hvis du har en god skærm, datasette etc., vil du kunne høre det ganske udemærket, hvorimod det vil lyde som båndssalat, hvis det ovennævnte udstyr ikke er af det bedste. Med mindre man er døv, eller har skruet helt ned for lyden, kan musikken sagtens genkendes. En styg rutine til din 64'er!

```

10 adr=53000
20 reada:ifa=-1then40
30 pokeadr+t,a:t=t+1:chk=chk+a:goto20
40 ifchk<>3865thenprint"der er en datafejl!":end
50 printchr$(147)chr$(18)" baand salat ":print:print"lavet af crep i 1987"
60 print:print"aktiveres med: sys 53000"
70 print:print"deaktiveres med tryk paa space":end
100 data 173,1,220,201,239,208,1,96,173,13,220,41,16,240,7
110 data 169,15,141,24,212,208,234,169,0,141,24,212,240,227,-1

```



**Program: O.K.**  
**Gevinst: Indoor Sports**  
**BWW-tekst 1**  
**Udstyr: C-64/128**

Er du træt af det evindelige »READY« på din 64'er? Det er denne nordmand åbenbart, for han har konstrueret et lille program, der omskifter »READY«'et med et »O.K.«!! En smart rutine, der gør det muligt, at gøre din hjemmecomputer lidt mere personlig!

Indsendt af:  
Helge Wunderlich  
Trolldalen  
4870 Fevik  
Norge

```
10 printchr$(147)"leser data ....."
20 f=49152:e=49200:rem f=start, e=end
30 iff>egoto80
40 c=f-int(f/256)*256:forf=ftof+7
50 readb:c=(c+b)and255:pokef,b:next
60 readb:ifb<>cthenprint"data error in";peek(63)+256*peek(64)
70 goto30
80 rem program continues here
90 printchr$(147):print" dette programmet forandrer 'ready.':print
100 print" til 'o.k.':print:print" efter reset eller run/stop restore,":print
110 print" kan programmet startes med":print:print" "chr$(18)" poke 1,54 "
120 sys49152
130 data 160, 32,162, 0,189, 0,160,157, 92
140 data 0,160,232,208,247,238, 6,192, 11
150 data 238, 9,192,136,208,238,169, 54,236
160 data 133, 1,162, 0,189, 41,192,157,131
170 data 120,163,232,224, 6,208,245, 96, 46
180 data 0, 79, 46, 75, 46, 32, 32, 0, 94
190 data 0, 0, 0, 0, 0, 0, 0, 0, 48
```

# RESET

Bøvse Reset giver sig selv - det er en reset beskyttelse. Hvis man prøver på at resette computeren, eller man trykker på RESTORE, ja så vil computeren skrive en skæg meddelelse på skærmen samtidig med, at den udstøder nogle bøvse-lyde!

Programmet er lavet af Allan Petersen, 13 år, som derudover ønsker at være anonym!

```
10 adr=32768
20 reada:ifa=-1then40
30 pokeadr+t,a:t=t+1:chk=chk+a:goto20
40 ifchk<>14814thenprint"der er en datafejl!":end
50 printchr$(147)chr$(18)"boevse reset er aktiveret!":end
100 data 10,128,10,128,195,194,205,56,48,0,120,76,14,128,169,7,141
110 data 5,212,169,133,141,6,212,169,15,141,24,212,169,0,162,17,141
120 data 5,212,142,4,212,141,134,2,32,68,229,162,0,189,111,128,157
130 data 66,5,232,224,33,208,245,169,10,141,33,208,141,32,208,162,0
140 data 232,224,255,208,251,162,0,232,224,9,208,251,169,2,141,33,208
150 data 141,32,208,162,0,232,224,255,208,251,162,0,232,224,9,208,251
160 data 173,18,208,141,1,212,76,58,128,19,15,18,18,25,32,9,3
170 data 45,18,21,14,32,23,1,19,32,8,5,18,5,32,2,5,6
180 data 15,18,5,32,25,15,21,33,-1
```

## Game Chars

Hvis man sidder og laver spil eller lignende, er det grimt at bruge de normale karakterer i 64'eren. Det ødelægger stemningen! Hvad gør man så? Jo, så kan man selv tegne karaktererne i en editor, eller man kan jo også bruge dette program. Det udstyrer nemlig din 64'er med nogle edderseje nye karakterer. Prøv selv!

```
program: Game Chars
```

```
10 clr:restore:adr=52000
20 fort=1to6:chk=0:fore=1to80
30 reada:ifa=-1then50
40 chk=chk+a:pokeadr,a:adr=adr+1:nexte
50 reada:ifa<>chkthenprint"fejl i linjerne "50+(t*50)"-"90+(t*50):end
60 nextt
70 printchr$(147)chr$(18)" game chars ":print:print"lavet af crep i 1987"
80 print:print"aktiveres med: sys 52000,1"
90 print:print"deaktiveres med: sys 52000,0":end
100 data 32,241,183,224,1,240,21,144,3,76,72,178,169,199,141,0
110 data 221,169,21,141,24,208,169,4,141,136,2,96,120,160,0,132
120 data 251,169,208,133,252,165,1,72,169,51,133,1,177,251,145,251
130 data 136,208,249,230,252,165,252,201,224,208,241,160,7,185,177,203
140 data 153,248,209,73,255,153,248,213,136,16,242,160,207,185,185,203,12081
150 data 153,8,208,73,255,153,8,212,136,192,255,208,240,160,79,185
160 data 137,204,153,128,209,73,255,153,128,212,136,16,242,160,15,185
170 data 217,204,153,8,209,73,255,153,8,213,136,16,242,104,133,1
180 data 169,148,141,0,221,169,5,141,24,208,169,192,141,136,2,88
190 data 96,62,39,7,28,56,0,56,0,62,39,103,254,238,238,238,10526
200 data 0,126,39,103,252,238,238,252,0,62,39,96,224,224,238,124
210 data 0,126,39,103,238,238,238,252,0,127,32,96,252,224,224,254
220 data 0,127,32,96,252,224,224,224,0,62,39,96,238,230,238,124
230 data 0,119,39,103,254,238,238,238,0,62,8,24,56,56,56,124
240 data 0,15,2,6,14,14,238,124,0,119,39,110,252,236,238,238,10164
250 data 0,112,32,96,224,224,224,254,0,99,39,127,238,238,238,238
260 data 0,103,39,119,254,254,238,230,0,62,39,103,238,238,238,124
270 data 0,126,39,103,252,224,224,224,0,62,39,103,238,238,124,30
280 data 0,126,39,103,252,238,238,238,0,62,39,96,124,14,238,124
290 data 0,127,8,24,56,56,56,56,0,119,39,103,238,238,238,124,10101
300 data 0,119,39,103,238,238,124,56,0,119,39,103,238,254,238,198
310 data 0,119,39,103,124,238,238,238,0,119,39,103,124,56,56,56
320 data 0,63,2,6,56,112,224,254,0,62,39,111,254,238,238,124
330 data 0,28,8,24,56,56,56,124,0,62,39,7,60,112,224,254
340 data 0,62,39,7,60,14,238,124,0,55,39,103,126,14,14,14,7560
350 data 0,127,39,96,124,14,238,124,0,62,39,96,252,238,238,124
360 data 0,127,39,14,56,56,56,56,0,62,39,103,124,238,238,124
370 data 0,62,39,103,126,14,238,124,0,28,8,24,56,56,0,56
380 data 0,119,51,102,0,0,0,0,0,-1,4349
```

## Verify

En rutine der mangler i mange MC-monitore. Den kan checke, om et areal er magen til et andet areal af samme størrelse. Det kan være nyttigt, hvis man f.eks. skal beskytte et program eller bare vil se, hvorvidt de to arealer er ens. Hvis man f.eks. vil se, om området fra 1024-2023 er identisk med området 49152-50151, skriver vi: sys50000 ,1024,4952,50151 (RETURN). I

dette tilfælde ser vi, om det normale skærmområde er magen til et andet ofte brugt skærmområde (jeg bruger tit skærmområdet 49152-50151, når jeg laver nye karaktersæt). Når du har kørt rutinen, kan du i adresse # 251 (\$fb) se om arealerne er ens eller ej. Hvis der står 0 er de ens, men hvis der står 1 er de ikke ens!

En rutine for dig, der pusler med 64'erenes bits og bytes!

```
10 adr=50000
20 reada:ifa=-1then40
30 pokeadr+t,a:t=t+1:chk=chk+a:goto20
40 ifchk<>10895thenprint"der er en datafejl!":end
50 printchr$(147)chr$(18)" ram verify ":print:print"lavet af crep i 1987"
60 print:print"aktiveres med:":print"sys 50000,omraade, fra,til":end
100 data 160,5,152,72,32,253,174,32,138,173,32,247,183,104,168,165,21
110 data 153,168,0,165,20,153,167,0,136,136,16,229,200,120,165,1,72
120 data 169,0,133,1,177,170,81,172,208,29,165,169,197,171,208,6,165
130 data 168,197,170,240,14,230,170,208,2,230,171,230,172,208,228,230,173
140 data 208,224,169,0,44,169,1,133,251,104,133,1,88,96,-1
```



## Disco Lys

Nu kan vi alle danse til råseje rytmer, mens vores 1541'er leverer lys-showet. Imponer vennerne, fortryl kæresten og lav fed fest! Start denne rutine og se din 1541'er levere lyset til dette års sejeste mega-party!

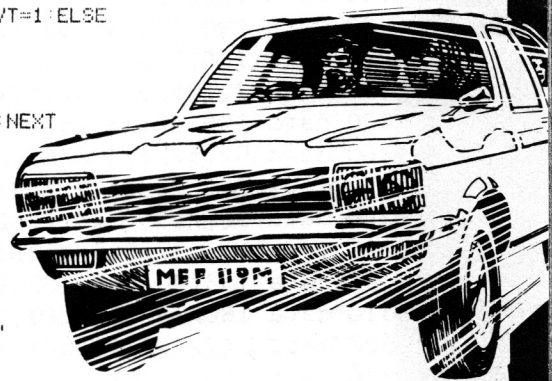
program: Disco Lys

```
10 adr=53000
20 reada:ifa=-1then40
30 pokeadr+t,a:t=t+1:chk=chk+a:goto20
40 ifchk<>17433thenprint"der er en datafejl!":end
50 printchr$(147)chr$(18)" disco lys ":print:print"lavet af crep i 1987"
60 print:print"aktiveres med: sys 53000"
70 print:print"deaktiveres ved at slukke":print"disk-drevet + sys 65511":end
100 data 160,50,169,8,32,177,255,169,111,32,147,255,169,77,32,168
110 data 255,169,45,32,168,255,169,87,32,168,255,152,32,168,255,169
120 data 5,32,168,255,169,1,32,168,255,185,85,207,32,168,255,32
130 data 174,255,136,16,205,169,8,32,177,255,169,111,32,147,255,169
140 data 85,32,168,255,169,67,32,168,255,32,174,255,96,169,254,170
150 data 32,27,5,32,27,5,202,224,1,208,248,32,27,5,32,27
160 data 5,232,224,255,208,248,240,235,138,72,73,255,168,169,248,141
170 data 0,28,202,208,248,169,240,141,0,28,136,208,248,104,170,96,-1
```

# Commodore T-16.

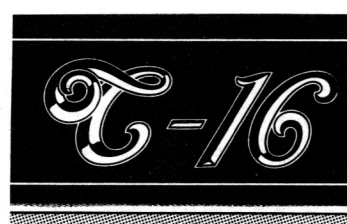


## CAR- race



```

10 COLOR0,1:COLOR4,1:DIMBA$(39)
20 PRINTCHR$(142)CHR$(8)
30 A=14552:POKE$2,55:POKE$6,55:SCNCLR
40 DATA2,00,8D,00,00,9D,00,38,8D,00
50 DATA D1,9D,00,39,E8,00,F1,60,8D
60 DATA FF,8D,7E,8D,E7,8D,18
70 FORT=1630T01647:READA$:POKET,DEC(A$)
80 NEXT:SYS1630:POKE65298,251:POKE65299,DEC("38")
90 FORT=14336T014343:READA$:POKET,DEC(A$)
100 NEXT:FORT=ATOR+15:READA$:POKET,DEC(A$):NEXT
110 CHAR1,16,2,"CAR RACE":CHAR1,12,4,"LAVET I 1987 AF:"
120 CHAR1,12,6,"CURT KJARSGAARD."
130 A$="":PRINTA$:INPUT"SVARHEDSGRAD 1-6":A$
140 A=VAL(A$):IFA<10RAD6THEN130
150 INPUT"V)D ELLER (T)ØR BANE.":A$:IFA$="V"THENVT=2:ELSEIFA$="T"THENVT=1:ELSE
GOTO150
160 LE$="":FORT=0TOR:LE$=LE$+" ":NEXT:IFVT=2THENCOLOR0,2
170 FORT=1T039:BA$(T)="":NEXT:LA=39-LEN(LE$):VOL8:POKE65286,23:SCNCLR
180 FORT=1TOLA:FORY=1TOT:BA$(T)=BA$(T)+" ":NEXT:BA$(T)=BA$(T)+"00"+LE$
190 BA$(T)=BA$(T)+"00":FORY=39TOT+LEN(LE$)STEP-1:BA$(T)=BA$(T)+" ":NEXT:NEXT
200 FORT=0T024:PRINTBA$(1):NEXT:BA=1:S=3074:POKE$0:TI$="000000"
210 VE=0:DATA3E,48,48,7E,48,48,4E,00,18,00,3C,66,7E,66,66,00
220 GETA$:IFA$=","THENVE=-1:ELSEIFA$="."THENVE=1
230 S=S+VE:GH=INT(RND(1)*3):IFGH=1ANDBA<LATHENBA=BA+1
240 IFGH=2ANDBA<1THENBA=BA-1
250 PRINTBA$(BA):IFPEEK(S)<32THEN270
260 POKES,0:ONVTGOTO210,220
270 FORT=100T0150:FORY=1T02:POKES,T: SOUND3,T*2,1
280 COLOR0,Y:NEXT:NEXT:SCNCLR:COLOR0,1
290 POKE239,0:CHAR1,15,11,"GAME OVER.":CHAR1,11,13,"DU KØRTE "+TI$+" KM"
300 GETKEY$:RUN
    
```

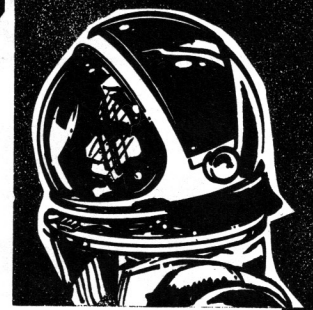


Denne rutine er en udbygning af OLD. Hvis du bruger en eller flere af mine andre rutiner i dit program vil den normale OLD blive slettet, men den nye SUPER OLD skal du blot load ind med .8.1 eller .1.1 (den sletter ikke nogle af mine andre rutiner) og den vil holde af sig selv. Når du har runnet programmet skal du, et par sekunder efter, trykke return og der vil nu blive savet et program der hedder SUPER OLD, load det for eftertiden. Hvis rutinen skal saves på bånd skal du skrive 17 i linie 30, istedetfor 8.

```

3 REM          * SUPER OLD *
1 REM
2 REM          AF CURT KJARSGAARD.
3 REM
10 FORT=819T0842:READA$:POKET,DEC(A$):B=B+DEC(A$):NEXT
20 IF B<1776THENPRINT"DATA FEJL":END
27 POKEDEC("0302"),DEC("33")
28 POKEDEC("0303"),DEC("03")
30 PRINT"Commodore"CHR$(34)"SUPER OLD"CHR$(34)".8.0302.034B8":MONITOR
1000 DATA9,01,8D,01,10,20,18,88
1010 DATA20,4E,88,09,12,8D,02,03
1020 DATA9,87,8D,03,03,4C,12,87
    
```

# W3VW45



## Darth Vader's Revenge

```
10 clr:restore:adr=49152
20 fort=1to20:chk=0:fore=1to90
30 reada:ifa=-1then50
40 chk=chk+a:pokeadr,a:adr=adr+1:nexte
50 reada:ifa<>chkthenprint"fejl i linjerne "50+(t*50) "-90+(t*50):end
60 nextt:sys49152
100 data 76,168,198,120,169,127,141,13,220,169,1,141,26,208,169,233,141,18
110 data 208,169,27,141,17,208,169,36,141,20,3,169,192,141,21,3,88,96
120 data 173,18,208,201,233,208,31,169,0,141,18,208,169,22,141,24,208,169
130 data 200,141,22,208,169,0,141,33,208,141,32,208,169,1,141,25,208,76,5
140 data 193,169,233,141,18,208,169,30,141,24,208,169,216,141,22,208,169,11115
150 data 0,141,32,208,141,33,208,169,1,141,25,208,230,162,32,159,255,173
160 data 141,2,13,137,198,240,3,76,5,193,173,89,198,141,0,208,173,16
170 data 208,41,254,13,90,198,141,16,208,173,0,220,41,4,208,29,173,90
180 data 198,208,7,173,89,198,201,25,144,17,56,173,89,198,233,2,141,89
190 data 198,173,90,198,233,0,141,90,198,173,0,220,41,8,208,29,173,90,10935
200 data 198,240,7,173,89,198,201,64,176,17,24,173,89,198,105,2,141,89
210 data 198,173,90,198,105,0,141,90,198,173,0,220,41,16,208,47,173,88
220 data 198,208,42,56,173,89,198,233,24,133,180,173,90,198,233,0,74,102
230 data 180,70,180,70,180,238,88,198,24,165,180,105,33,133,251,133,253,169
240 data 7,105,0,133,252,105,212,133,254,104,168,104,170,104,64,32,3,192,11539
250 data 169,4,133,252,160,0,185,226,196,153,0,56,200,192,8,208,245,160
260 data 0,152,153,0,57,200,192,8,208,248,160,0,185,234,196,153,0,58
270 data 200,208,247,169,232,141,248,7,160,7,169,233,153,248,7,136,208,250
280 data 169,255,141,28,208,169,0,141,29,208,141,23,208,141,16,208,169,3
290 data 141,37,208,169,8,141,38,208,169,212,141,1,208,160,0,185,236,197,12490
300 data 153,40,208,200,192,7,208,245,169,6,141,39,208,169,147,32,210,255
310 data 160,39,169,160,153,152,7,169,5,153,152,219,136,16,243,160,5,162
320 data 24,24,32,240,255,160,0,185,244,197,240,6,32,210,255,200,208,245
330 data 160,0,162,23,24,32,240,255,160,0,185,25,198,240,6,32,210,255
340 data 200,208,245,169,4,141,35,208,169,14,141,36,208,160,24,169,0,153,12137
350 data 0,212,136,16,250,169,255,141,15,212,169,128,141,18,212,169,143,141
360 data 24,212,169,15,141,139,198,169,3,141,136,198,169,0,141,88,198,141
370 data 137,198,170,142,90,198,169,184,141,89,198,138,10,168,10,10,10,10
380 data 24,105,31,153,91,198,169,0,153,92,198,153,3,208,169,60,157,105,198
390 data 32,186,195,232,224,7,208,222,169,255,141,21,208,173,30,208,173,11832
400 data 31,208,173,30,208,41,1,240,3,76,230,195,173,141,2,208,251,32
410 data 228,255,201,136,208,9,169,32,160,0,145,251,76,188,196,173,88,198
420 data 208,3,76,27,195,160,0,169,32,145,251,56,165,251,233,40,133,251
430 data 133,253,165,252,233,0,133,252,24,105,212,133,254,173,27,212,9,8,145
440 data 253,169,0,145,251,173,31,208,41,254,240,103,133,167,141,138,198,12719
450 data 162,0,70,167,70,167,144,71,169,32,141,5,212,169,246,141,6,212
460 data 169,129,141,4,212,169,233,157,249,7,160,10,173,236,197,157,40,208
470 data 140,1,212,165,162,197,162,240,252,136,208,238,189,236,197,157,40,208
480 data 169,233,157,249,7,169,128,141,4,212,138,72,189,105,198,73,255,74
490 data 74,74,32,201,196,104,170,232,224,7,208,176,173,138,198,73,255,45,13128
500 data 21,208,141,21,208,76,226,194,238,88,198,173,88,198,201,21,208,57
510 data 160,0,140,88,198,169,32,145,251,173,31,208,173,30,208,173,21,208
520 data 41,254,208,35,173,139,198,240,3,206,139,198,160,38,162,23,24,32
530 data 240,255,173,139,198,73,15,170,169,0,32,205,189,169,100,32,201,196,76
540 data 224,193,174,139,198,160,0,200,208,253,202,208,250,138,10,168,185,12788
550 data 91,198,153,2,208,189,105,198,153,3,208,185,92,198,133,168,56,138
560 data 168,200,200,169,0,42,136,208,252,133,167,73,255,45,16,208,164,168
570 data 240,2,5,167,141,16,208,232,224,7,208,204,162,0,138,10,168,189
580 data 105,198,24,125,112,198,201,210,176,4,201,50,176,6,32,186,195,76,178
590 data 195,157,105,198,24,185,91,198,121,120,198,133,167,185,92,198,121,12344
```

# Nummer to!

**Program: Darth Vaders Revenge**

**Gevinst: Indoor Sports  
Boulder Dash Constr. Kit  
BMW-tekst 1  
Amiga Gold-disk**

**Udstyr: C-64/128**

Darth Vaders er ham den sort-klædte vandal vi kender fra Star Wars. Han vender her frygteligt tilbage i Darth Waders Revenge... Pløk alle fjenderne og undgå at blive ramt! Andet er der ikke at sige om dette seje spil, som gled lige ind på en flot andenplads.



600 data 121,198,133,168,208,14,165,167,201,31,240,2,176,6,32,186,195,76  
610 data 178,195,165,168,240,12,165,167,201,64,144,6,32,186,195,76,178,195  
620 data 165,167,153,91,198,165,168,153,92,198,232,224,7,208,165,76,30,194  
630 data 134,169,132,170,173,27,212,41,5,170,189,70,198,166,169,157,112,198  
640 data 173,27,212,41,5,10,168,185,76,198,166,170,157,120,198,185,77,198,12430  
650 data 157,121,198,164,170,166,169,96,169,235,141,248,7,169,1,141,137,198  
660 data 169,9,141,5,212,169,160,141,6,212,169,33,141,4,212,162,100,142  
670 data 1,212,160,0,173,27,212,141,39,208,141,0,212,136,208,244,202,208  
680 data 236,169,234,141,248,7,169,0,141,29,208,141,23,208,169,32,141,4  
690 data 212,169,168,141,6,212,169,129,141,4,212,162,100,142,1,212,160,0,12038  
700 data 140,0,212,173,27,212,141,39,208,136,208,244,202,208,236,169,232,141  
710 data 248,7,169,6,141,39,208,169,0,141,29,208,141,23,208,169,128,141  
720 data 4,212,162,100,160,0,136,208,253,202,208,250,169,0,141,137,198,168  
730 data 153,2,208,200,192,14,208,248,141,16,208,160,0,169,32,145,251,173  
740 data 30,208,206,158,7,206,136,198,173,136,198,240,3,76,224,193,160,0,12883  
750 data 185,203,7,217,222,7,240,5,176,11,76,178,196,200,192,6,208,238  
760 data 76,178,196,160,6,185,202,7,153,221,7,136,208,247,32,159,255,32  
770 data 228,255,201,136,208,246,160,6,169,48,153,202,7,136,208,250,76,158  
780 data 193,170,160,6,56,185,202,7,105,0,201,58,144,2,169,48,153,202  
790 data 7,136,208,239,202,208,233,96,48,16,32,48,16,32,48,16,0,0,11520  
800 data 0,0,20,0,0,20,0,0,20,0,0,20,0,16,60,4,16,255  
810 data 4,19,235,196,31,235,244,63,190,252,63,190,252,254,255,191,254,255  
820 data 191,251,255,239,251,255,239,239,255,251,239,255,251,63,255,252,1,0  
830 data 64,5,65,80,21,85,84,0,0,170,0,2,170,128,10,170,160,42,170,168  
840 data 41,105,104,169,105,106,169,105,106,170,170,170,170,150,170,186,11146  
850 data 170,174,190,170,190,175,255,250,171,255,234,170,170,170,42,170,168,42  
860 data 170,168,10,170,160,14,170,176,12,170,48,12,0,48,60,0,60,0  
870 data 128,32,8,32,10,32,0,168,160,34,162,160,138,168,130,162,197,42  
880 data 40,115,8,131,190,224,46,188,162,11,127,64,130,201,88,10,186,74  
890 data 35,190,232,0,141,32,10,38,10,38,166,160,0,168,168,130,138,40,10068  
900 data 10,3,128,8,32,0,0,2,32,0,0,0,0,0,20,0,0,20  
910 data 0,0,20,0,0,20,0,16,60,4,16,255,4,19,235,196,31,235  
920 data 244,63,190,252,63,190,252,254,255,191,254,255,191,251,255,239,251,255  
930 data 239,239,255,251,239,255,251,63,255,252,1,0,64,5,65,80,21,85  
940 data 84,0,0,13,6,4,1,7,8,12,13,14,8,152,211,67,79,82,8652  
950 data 69,58,158,48,48,48,48,48,48,48,32,32,152,200,73,71,72,32,211  
960 data 67,79,82,69,58,158,48,48,48,48,48,48,0,129,204,73,86,69  
970 data 83,58,51,32,32,5,196,193,210,212,200,32,214,193,196,197,210,39  
980 data 211,32,210,197,214,197,206,199,197,32,32,129,204,69,86,69,76,58  
990 data 48,32,32,0,1,255,2,254,3,253,1,0,255,255,2,0,254,255,9483  
1000 data 3,0,253,255,0,184,0,142,0,100,0,60,0,223,0,196,0,154  
1010 data 0,4,1,97,97,81,84,171,171,97,1,1,1,1,3,3,1,0  
1020 data 3,0,1,0,255,255,3,0,1,0,255,255,1,0,0,0,3,0  
1030 data 4,15,153,147,14,153,208,82,69,83,83,32,5,70,55,32,153,84  
1040 data 79,32,40,82,69,41,83,84,65,82,84,46,169,0,141,32,208,141,6336  
1050 data 33,208,162,0,189,141,198,32,210,255,232,224,27,208,245,165,197,201  
1060 data 3,208,250,76,11,193,-1,3668

## Quote Escape

hvis man f.eks. sidder og skriver noget i en »print«-sætning, hvor man har sat gåseøjnene til, kan det være irriterende hvis man begår en fejl! Man kan ikke bruge cursor-knapperne til at rykke cursoren tilbage for at rette det - der kommer bare nogle underlige tegn! Hvis du bruger denne rutine kan

du undgå, at spille tid på noget så banalt som at skulle trykke SHIFT + RETURN for derefter at gå op på linjen igen og rette fejlen. Med denne rutine skal du hvergang det nævnte problem opstår, bare trykke på f1-knappen og vupti! Du kan nu gå frem og tilbage med cursoren, som om intet var hændt!

```
10 adr=940
20 reada:ifa=-1then40
30 pokeadr+t,a:t=t+1:chk=chk+a:goto20
40 ifchk<>10737thenprint"der er en datafejl!":end
50 printchr$(147)chr$(18)" quote escape ":print:print"lavet af crep i 1987"
60 print:print"aktiveres med: sys 940,1"
70 print:print"deaktiveres med: sys 940,0":end
100 data 32,241,183,224,1,240,25,144,3,76,72,178,162,72,160
110 data 235,173,230,3,240,4,166,251,164,252,142,143,2,140,144
120 data 2,96,173,143,2,133,251,173,144,2,133,252,169,1,141
130 data 230,3,169,231,141,143,2,169,3,141,144,2,96,0,165
140 data 203,201,4,240,3,76,72,235,169,64,133,203,169,0,133
150 data 199,133,212,133,216,76,72,235,-1
```

---

### Program: Sprite Color Scroll

Gevinst: Indoor Sports  
BWW-tekst 1

Udstyr: C-64/128

```
5 poke53280,0:poke53281,0
10 reada:ifa=-1then20
15 poke32000+i,a:i=i+1:goto10
20 printchr$(147)chr$(158)" start: sys32000"
25 print" poke251,y-reg. for sprites."
30 print" poke252,antal farver."
35 print" poke32100,fart:p0253,255."
40 goto90
45 data169,130,133,251,169,27,133,252,169,255,133,253,120,169,127,141,13,220
50 data169,1,141,25,208,141,26,208,162,38,160,125,142,20,3,140,21,3,88,96,169
55 data1,141,25,208,160,0,166,251,142,1,208,142,3,208,142,5,208,142,7,208,142
60 data9,208,142,11,208,142,13,208,142,15,208,185,131,125,236,18,208,208,251
65 data141,39,208,141,40,208,141,41,208,232,232,200,196,252,208,232,230,253
70 data165,253,201,2,208,25,169,255,133,253,166,252,189,131,125,72,189,131
75 data125,157,132,125,202,224,255,208,245,104,141,131,125,76,49,234
80 rem **** farver ****
85 data6,14,3,14,6,0,0,2,10,1,10,2,0,0,9,8,7,8,9,0,0,5,13,1,13,5,0,0,-1
90 rem **** demo ***
95 poke53269,7:poke53248,120:poke53250,160:poke53252,200:poke251,0
100 fort=0to63:poke832+t,255:next:poke2040,13:poke2041,13:poke2042,13:sys32000
105 fori=0to230:poke251,i:next:fori=230to128step-1:poke251,i:next
110 poke53271,7:poke53277,7:poke53275,7:printchr$(19)chr$(144)
115 fort=0to9:print:next:print" lavet af stig"
120 print:print" for ic-run."chr$(158)
```

Navnet, Sprite Color Scroll, forklarer egentlig udmærket i sig selv, hvad dette program kan præsentere. Du kan lave nogle seje effect's i dine sprites - du kan nemlig få et ton farver til at rulle glat ned igennem dem! Det ser bare sååå syret ud!!!

Et absolut prisværdigt program lavet af en gæv gut, ved navn Stig Norre.



# OT-beskytter

## Nummer tre!

**Program:** OT-Beskytter II  
**Gevinst:** Boulder Dash Constr. Kit  
**BWW-tekst 1**  
**Udstyr:** C-64/128 + Diskstation

Dette program, er ganske simpelt en LIST-beskyttelse... Nej! Ikke bladet... Første gang man list'er det beskyttede program, vil der blive udskrevet en morsom besked på skærmen. Programmet vil ikke bive list'et! Når det samme program så har været runnet og er stoppet, vil det, hvis man list'er det igen, udføre en cold-reset. D.v.s. at computeren resetter via software!

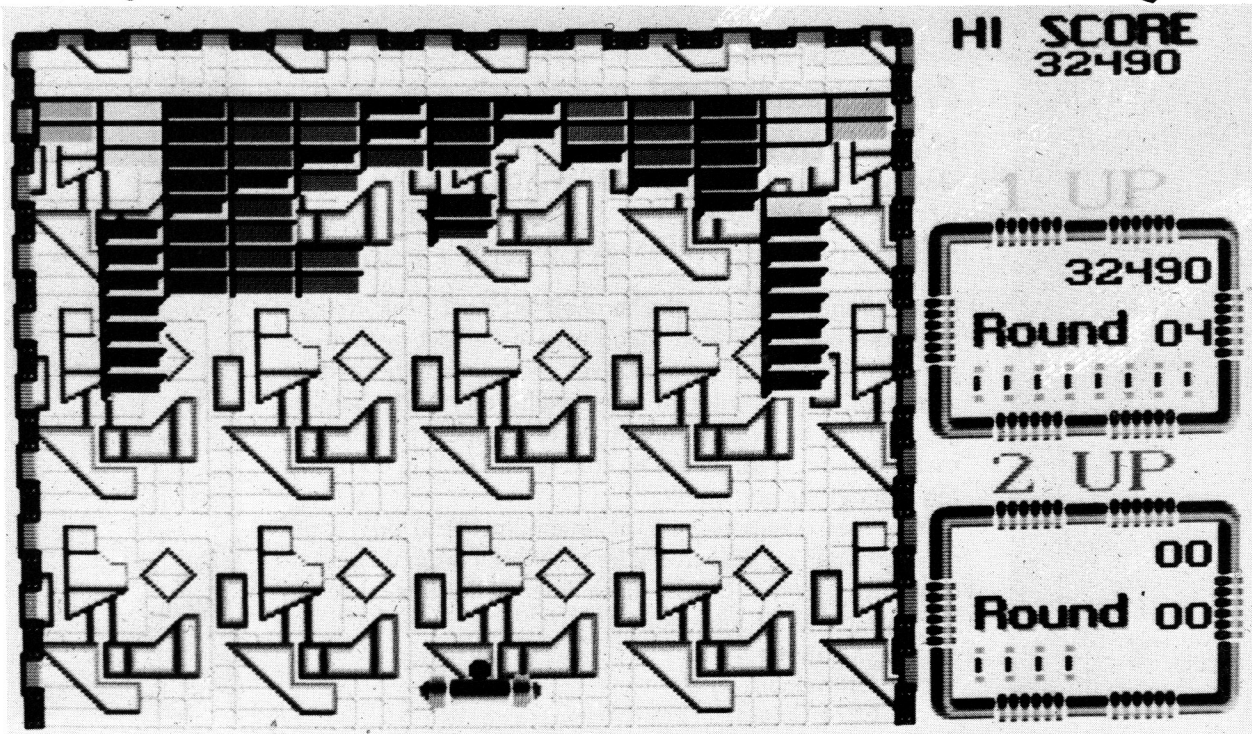
Man skal dog huske på, at linie-numrene 0 og 1 i det program man vil beskytte, ikke må være i brug, da beskyttelses programmet bruger disse linier til lidt fiksfakserier!

Programmet er indsendt af:

Ole Tange  
Ommegårdvej 8  
6971 Spjald

### OT-Beskytter II

```
10 printchr$(147):input"indtast programnavn";a$
20 open15,8,15,"i0":gosub100:open4,8,4,"0:""+a$+"",p,r":gosub100
40 open5,8,5,""0:""+a$+"",p,w":gosub100:goto150
100 ::input#15,er,er$,t,s:ifer<>0thenprinter,er$,t,s:end
102 return
110 ::get#4,a$:a$=left$(a$+chr$(0),1):print#5,a$::return
120 ::get#4,a$:a$=left$(a$+chr$(0),1):a=asc(a$):return
130 ::print#5,a$::return
150 gosub110:gosub110:gosub1000:gosub120:z=z+248
155 z=z+5:x=a:gosub120:x=x+a*256:x=x+z:y=int(x/256):x=x-y*256:a$=chr$(x)
160 gosub130
170 a$=chr$(y):gosub130:gosub110:gosub110
175 a$=chr$(0)+chr$(58)+chr$(58)+chr$(58)+chr$(58):gosub130
180 gosub110:ifa$<>chr$(0)then180
185 gosub120:f1=0
190 ifa=0thengosub120:f1=1:ifa=0thena$=chr$(0)+chr$(0):gosub130:goto999
193 iff1=1thena$=chr$(0):gosub130
195 goto155
999 forh=1to255:closeh:next:end
1000 data175,8,255,255,143,34,141,147,5,18,32,32,32,32,32,32,32,32
1002 data78,65,65,44,32,68,85,32,86,73,76,76,69,32,78,79,75,32,76,73,83,84,69
1004 data32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32
1006 data80,82,79,71,82,65,77,77,69,84,44,32,72,86,65,68,32,63,63
1008 data32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32
1010 data68,69,83,86,65,69,82,69,32,69,82,32,68,69,84,32,66,69,83,75,89,84,84
1012 data69,84,32,77,69,68,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32
1014 data79,84,45,66,69,83,75,89,84,84,69,82,32,73,73
1016 data32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,146,34,204,0
1020 data253,8,0,0,0,58,58,58,58,151,55,55,52,44,50,53,50,58,151,55
1022 data55,53,44,50,53,50,58,151,56,49,56,44,50,53,50,58,151,56,49,57
1024 data44,50,53,50,58,151,55,56,56,44,53,50,58,151,56,48,56,44,50,50
1026 data53,58,151,52,51,44,49,55,53,58,151,52,52,44,48,48,56,0
2000 fort=1to252:readwe:a$=chr$(we):gosub130:next:return
```



### Starten

Starten er oftest den mest deprimierende fase. Man kan ikke finde ud af, hvilken slags spil man skal lave. Man overvejer både det ene og det andet - platform-, arcventure-, adventure og shoot'em up-spil! Efter utallige overvejelser og spil-fyldte mareridt, ender man som oftest med, at vælge at lave et shoot'em up. Det er uden tvivl det mest anvendte spilkoncept og desuden også det nemmeste at lave. Det kan være meget godt, men der findes jo i forvejen en lang række titler af den slags spil på markedet. Mange af dem er konverteringer og licenser som oftest viser sig, at være noget værre crap!

Ikke desto mindre vil statistikken også vise, at disse som regel er de spil der ligger lunest på hitlisterne. Så tænker man at »Jamen, så kontakter jeg da bare Ocean og får dem til at lade mig lave en af deres licens-titler! Næh nej, sådan kan man ikke gribe det an. Man må selv være original og starte sin karriere med at lave et spil folk lægger mærke til. Så kan man

endelig tillade sig at kontakte de stærke firmaer og prøve at få hevet nogle seje kontrakter hjem.

Lad os koncentrere os om konceptet Shoot'em Up, og droppe vore drømme om, at vi konverterer en titel for Ocean.

### Gameplay

Man må først og fremmest finde ud af, hvordan handlingen i spillet skal være. Skal det være med eller uden scroll effekter. Skal det være originalt og avanceret, eller skal det være et ganske almindeligt shoot'em up. Det mest nærliggende er nok, at lave et originalt og avanceret spil, men det vil vi let og elegant hoppe over. Det er alt for kompliceret for nybegyndere. Vi vil meget hellere lave et nogenlunde simpelt, scrollede shoot'em up.

Til dette formål, kunne man måske nedlade sig til, at bruge det meget omtalte Shoot'em Up Construction Kit. Det har nogle fede features og er lettilgængeligt, men gerne vil have den, man kan ikke bestemme hvordan kontrolpanelet skal se ud, og

det færdige program fylder 249 blokke! Irriterende! Endnu mere irriterende er det, at man ikke kan lave et stykke musik, som kan præsentere spillet.

Man må dog ikke overse, at du med SEUCK kan have utallige sprites på skærmen ad gangen og du kan hurtigt designe din baggrund. Ydermere, kan du selv lynhurtigt bikse nogle seje lyd-effekter sammen, og du slipper for at løse en hulens masse problemer, som du uden diskussion ville komme ud for, hvis du selv stod for programmeringen. I enkelte tilfælde kan SEUCK måske vise sig at være et kærkomment alternativ, men vil man lave et lækkert spil, og selv bestemme hvordan og hvorledes, så er der ikke andet for, end at sætte sig ned og lave det hele selv!

Et shoot'em up går hovedsageligt ud på, at skyde så mange modstandere som overhovedet muligt. Eagles, MACH, Tiger Mission og hvad de nu ellers hedder, er deciderede Shoot'em up's. Her skal du skyde, skyde og atter skyde. Modsat adventurespil, hvor du normalt ikke

lige frem skal skyde, men hvor du højst sandsynligt kommer ud for, at du skal dræbe en del modstandere. I adventure's kan du overveje dine bevægelser før du gør dem. Det har du ikke tid til i et shoot'em up, hvor du nærmest spiller pr. intuition. Derfor er det blevet både anerkendt og accepteret, at handlingen i et shoot'em up, hovedsageligt går ud på, at eliminere så mange fjender som muligt for at gennemføre. I spil af den type, kan næsten enhver computerejer forstå, hvad der foregår. Sandsynligvis er det derfor, lige netop den slags spil bliver solgt i større mængder end andre. Det gør det attraktivt, at producere et spil, hvor handlingen udelukkende går ud på enn ting - at skyde...

Man behøver ikke, at benytte sig af scrollede baner, men det er helt sikkert sejere, hvis man har en bunke baggrundsgrafik, der kører glat op, ned eller sidelæns hen ad skærmen. Så kan man eventuelt lave det sådan, at angrebsformationerne er afhængige af, hvilket stykke af banen man befinder sig over.

# \* Hvor'n man laver games! \*

**Vil man lave et spil, der skal opnå international anerkendelse og succes, er der mange faktorer man skal tage hensyn til. Man skal fra starten gøre sig det klart, bl.a. hvordan gameplayet skal være, hvordan grafikken skal opbygges og hvordan og hvorledes udformningen af spillet skal foregå. Det har vi fået en professionel spilprogrammør til at gå i dybden med.**

**Bo. R. Petersen**

Den teknik benytter bl.a. MACH, Slap Fight og Tiger Mission sig af. Man kan selvfølgelig også bare lave enhedsformationer, hvor angrebene finder sted i en systematisk rækkefølge, som man har defineret via en table. Det betyder, at angrebene er uafhængige af, hvor på skærmen man befinder sig og, at de altid vil være i samme rækkefølge, uanset hvor langt man nu er kommet i spillet.

Nu har man jo, med en computer, utallige muligheder for, lave spillet som man selv synes det skal være. Derfor skal man måske ikke kigge på gamle ideer, men simpelthen selv udvikle nye, og prøve at fuldføre dem.

## **Grafik**

Den væsentligste hovedingrediens i et shoot'em up, er grafik'en. Det er også hard work - man skal lave opstartsbillede, baggrundsgrafik, tekstkarakterer og sprites.

Opstartsbilledet tegner man nok hyppigst i højopløsning. Det kan man gøre ved hjælp af programmer som

Koala Painter, Art Studio og diverse andre. Her spiller hukommelsesproblemet normalt ikke ind, da opstartsbilleder normalt vises i et lille forprogram, som bliver loadet ind før selve spillet. Det er selvfølgelig op til en selv, hvordan man vil gøre det, men vælger man, at have det samme med selve spilsektionen, bliver man sikkert nødt til, at skære ned på selve spillet. Det ville jo være en skam, da man bør gøre det til en regel, at benytte så meget af hukommelsen som muligt udelukkende til, at gøre spillet det bedre.

Baggrundsgrafikken skal tegnes i multi-colour i en font-editor, så det ikke fylder mere end højst nødvendigt. En grafikblok fylder 2 Kbytes (2048 bytes) fordelt på 256 enkelte karakterer. Man designer denne grafik ud fra den forudsætning, at der under ingen omstændigheder må være at finde, 2 identiske karakterer i hele grafik-set'et. Hvis man følger dette, vil man kunne mixe de forskellige figurer og objekter, ved hjælp af de karakterer man har tegnet i font-editor'en. I et

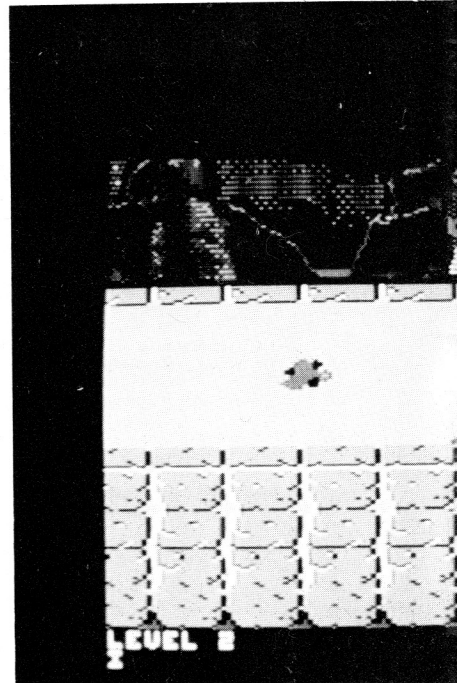
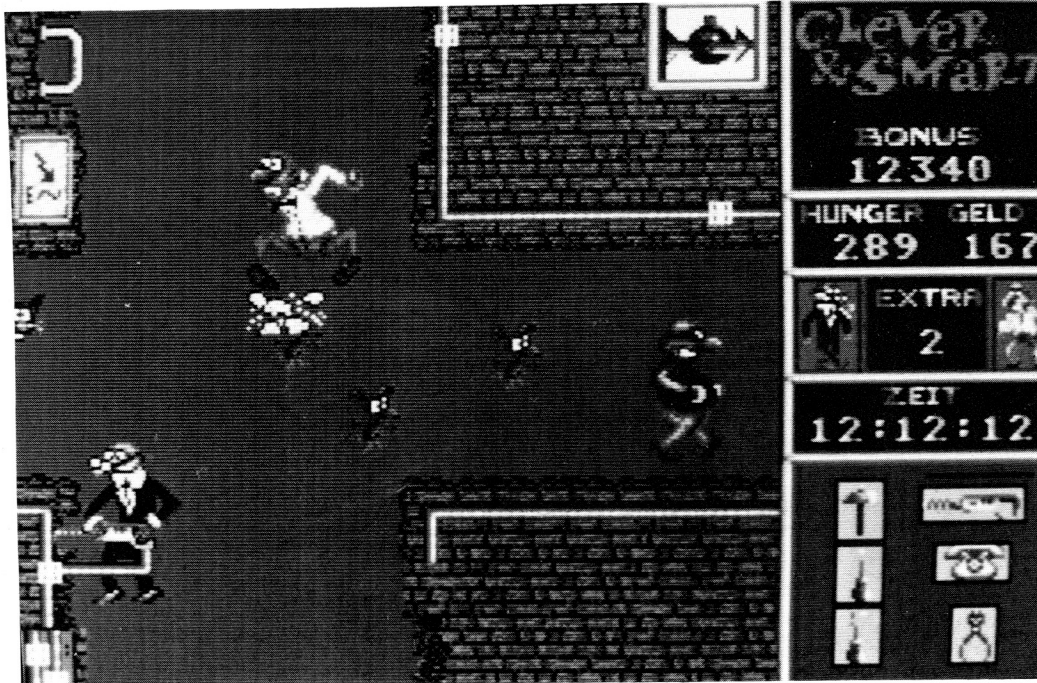
shoot'em up, er der absolut ingen regler for, hvordan grafikken skal se ud. Der er en af fordelene, da det er meget besværligt, at få virkelige ting tegnet pænt på en lille computer som 64'eren. Derfor, kan man lave sin grafik så fantasifuld som man nu ønsker det.

Ikke nok med, at baggrundsgrafikken skal tegnes. Der skal også tegnes et ton baner. Det gør man ved, at man i en bane-editor (du kan nemt lave en selv) bygger de enkelte objekter ud fra de grafik-karakterer man har tegnet i en font-editor. Sådan kan man tegne banen på en nem og overskuelig måde. For at spare hukommelse (Kilobytes, Bytes, osv.) og tid, kan man evt. bygge banen således, at man i stedet for hele sammensatte klodser, bare sætter et enkelt tegn, som så symboliserer vedkommende objekt. Når så banen skal skrives ud på skærmen, læser man dette tegn, og ud fra en tabel over hvordan karaktererne skal sættes sammen, opbygger man objektet og plotter det ud på skærmen. Første eksempel er betyde-

ligt nemmere at have med at gøre, men hvis der skal være mange baner, kan det tit gå hen og tage en masse tid, og det bliver et stort problem. Derfor benytter mange sig af det andet eksempel med tegn-symboler.

At tegne tekstkarakterer er såmænd ganske simpelt. Det er jo de tegn man ser på skærmen, når man sidder og skriver. Til dette formål, skal man også bruge en font-editor, men denne gang i single-colour. Man kan enten omdefinere de karaktertegn, der bliver brugt af computeren, eller man kan vælge at tegne nogle splinternye tekstkarakterer. Man skal måske også tegne nogle nye tegn, med det formål for øje, at bruge dem til illustrationer osv. Bla. kan man jo tegne en flot stjerne, nogle pile eller andet nyttigt og interessant tingel-tangel.

Sprites derimod, er betydeligt sværere at tegne. Her skal man jo ikke kun tegne een enkelt figur, men ca. 8-10 forskellige figurer for hvert spriteobjekt. Det hænger sammen med, at en animeret



sprite selvfølgelig skal animeres. Det gøres ved, at man placerer udgaver af grundfiguren set fra forskellige vinkler, i de antal sprites det nu kræver, at designe animationen. D.v.s., at man først tegner en såkaldt grundfigur. Det er selve spritefiguren, som man går ud fra, når animationen skal gennemføres. Det næste man gør, er at man tager grundfiguren - den første sprite - og ændrer lidt på dens udseende, så det ser ud som om, at man ser den fra en anden synsvinkel. Sådan bliver man ved, men istedet for, i de næste figurer, at gå ud fra grundfiguren, så går man ud fra den forudgående spritefigur. Det gør man indtil man kommer tilbage til den oprindelige figur (grundfiguren).

Når alt dette er gjort, og man har tegnet en masse forskellige figurer, så kan man i selve spillet, bare køre en lille sekvens inde i en spriteblock, som svarer til den pågældende grundfigurs forskellige figurer. Den eneste umiddelbare grund til, at man skulle følge dette eksempel, er at det jo ikke ser særligt spændende ud med

en lille grim figur, der slet ikke bevæger sig. Prøv f.eks. at sammenligne et stort købespil med de spil, vi har udlistet her i bladet! For at kunne tegne nogle pæne animerede spirter, bør man benytte et program som f.eks. Sprite-Machine, der har en masse lækre indbyggede kommandoer.

### Features

Spillet bør have en masse smarte detaljer indbygget. Det bør det, fordi spilleren f.eks. på et tidspunkt kan gå hen og blive træt af, at der spilles en melodi i baggrunden mens han prøver på at koncentrere sig om selve spillet. Sådanne detaljer kan måske være sound on / off, definering af styringsmetode, mulighed for at pause spillet, musik on / off, og en lille demonstration, en demo, af selve game't, så man kan danne sig et billede af handlingen, før man går igang.

Derudover, skal man sikre sig, at spillet er blevet ordentligt dokumenteret. Det kan man gøre ved, enten at lave en indbygget instruktions sekvens, eller en trykt instruktions manual. Det er meget

vigtigt, for samtidig med, at det giver en god præsentation, så giver det også spilleren mulighed for, at få et lille indblik i hvad han skal gøre hvor i selve spillet.

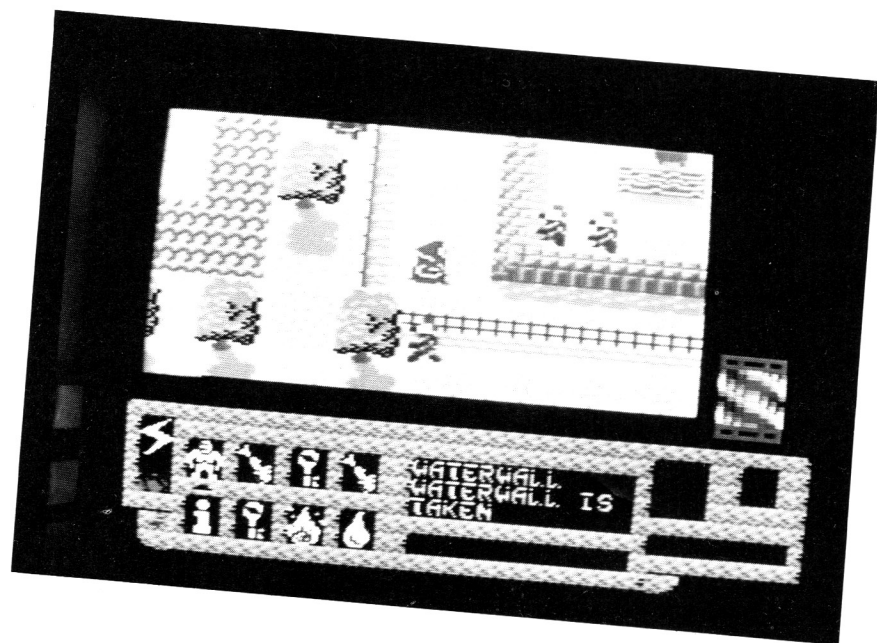
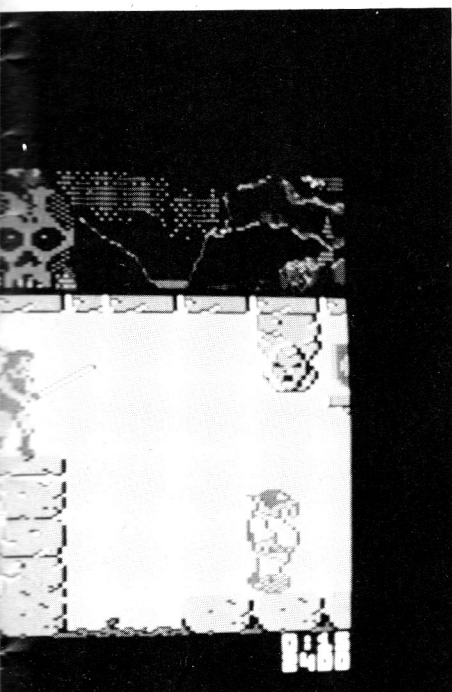
Under Features, hører også selve spillets kvalitet. Kvalitet er et begreb, der er meget svært at definere. Man kan ikke umiddelbart gå hen og sige, hvorvidt et vilkårligt spil er af go'kvali' eller ej! Derfor skal man satse på, at gøre alle udførelserne så glatte som muligt. Derved mener jeg, at man bl.a. skal benytte sig af raster-interrupts. Spritene skal rykke mens raster'en er uden for skærmområdet, og ligeså skal banen. Så undgår man, at skærbilledet lige pludseligt begynder at ryste og vibrere. Hvis f.eks. banen og spritene kun befinder sig i rasterområdet 50-200, så skal man ikke begynde at rokere rundt på nogle af delene, før man er kommet uden for dette område, til rasterlinie 200 eller 201. Samtidig skal man sørge for, at alle de nødvendige rutiner er blevet udført, før rasterlinien kommer tilbage til linie 50.

Et smart trick, som mange programmører ignorerer, er

at man kan vælge at lade kun en del af programmet blive udført i selve interruptet. Det kan bl.a. være skærmopdeling og de rutiner, der hele tiden skal gennemføres. Resten af programdelen udfører man uden for interruptet. Man laver en rutine, der hele tiden venter på, at raster'en skal komme til linie 200 (for at bruge det forrige eks.). Når så raster'en er på linie 200, så udfører man den del af programmet, der nu skal udføres. Det giver en hel del mere flexibilitet, idet man ikke behøver at køre denne rutine igennem, hvergang interruptet aktiveres. Man kan f.eks. lade den hoppe til en anden rutine der skal udføres i næste interrupt, for derefter at lade den hoppe tilbage.

### Lyd/Musik

En ting, der hører shoot'em up's til, er seje lydeffekter og tjekkede melodier! Mange kan ikke selv finde rundt i de mange forskellige lydregistre og toner. Derfor er der på markedet, en håndfuld mere eller mindre udelige musikprogrammer. Lad mig advare jer mod disse! Lyden bliver ofte direkte elendig, og slut-



udgaven af musikken fylder omkring de 50 blokke!!! Det er alt for høj en pris at betale - folk gider ikke købe spil med dårlig lyd, og lyden bruger alt for meget hukommelse. Derfor er der opstået en lille flok af musikprogrammører. Deriblandt kan nævnes navne som Rob Hubbard, Martin Galway, Fred Gray, og ikke mindst den danske musikvirtuose Johannes Bjerregaard. Den sidstnævnte har kaldt sit firma Danish Music Company, og han har leveret lyd siden til bl.a. Tiger Mission, Eagles og MACH.

Han har fået skabt sig et godt ry, så istedet for, at man går ud og kontakter en fyr, der kalder sig musikprogrammør, så skal man kontakte Johannes Bjerregaard! Han kan, i modsætning til diverse andre danske musikprogrammører, prale af, ikke at bruge klodsede købeprogrammer - han leverer den ægte vare. Så vil man give sit spil et ekstra skub på vejen til succes, bør man afgjort få J.B. til, at levere musikken til sit spil. Husk på, det er vigtigt med et godt lydsortiment, for anmelderne og spillerne er meget kræsne

m.h.t. computermusik og lydeffekter!

#### **Spilbarhed**

Spillet skal være spændende og lettilgængeligt. Det skal være let og simpelt, at starte i spillet, og spilleren skal fra starten have et godt indtryk af, hvor stort og godt spillet er. Med lettilgængeligt mener jeg, at det skal være nemt at finde rundt i spillet, de forskellige sektioner og opstartsmenuerne. Det er ikke nogen god ide, at lave store komplicerede indledninger, som spilleren alligevel ikke forstår - og iøvrigt ikke gider spille tid på! Man skal f.eks. ikke lave en masse menuer, hvor man kan vælge dit og dat. Det skal man indbygge i startskærmen. Iøvrigt, bør startskærmen ikke ændre udseende hele tiden, hvergang en ny effekt skal udføres på denne. Hi-score listen skal helst kunne kombineres med startskærmen, så billedet ikke differerer for meget. Det virker både forvirrende og uprofessionelt, og desuden er det s... irriterende.

Spændende skal det være, i form af flotte introduktioner, glatte overgange mellem de forskellige sektioner, og

effektfulde detaljer som bl.a., at spille en tone eller et musikstykke, når man går fra startskærmen og over i spillet. Under spændende hører også, at spillet på intet tidspunkt må gå hen og blive kedeligt, får så får spilleren et dårligt indtryk af det, og rygter spredes jo hurtigt. Spillet skal præsenteres ordentligt, og det bliver det jo ved, at man laver en spændende startskærm og en styg hi-score list. Man skal også lave en lille animation ligefør man gør videre til spillet. Det kan man lave, som i f.eks. Uridium, hvor der flyver et stort moderfartøj ind på skærmen, og smider spilleren ud i en hårrejsende space-fight!

**Til konstruktionen af et spil, kan nævnes mange andre detaljer som turbo-load og beskyttelse. Spillet bør loades hurtigt, ved hjælp af en turbo, hvadenten det er på bånd eller disk. Der er ingen, der gider at sidde og vente nat og dag på, at et spil skal loades ind. Man kan jo sammenligne det med Fast-Food restauranterne, hvor folk ikke gider vente længere end højst 3 mi-**

**nutter på, at få serveret deres mad. Turbo'en kan man enten købe hos diverse fabrikanter - så skal man betale royalty, eller man kan selv konstruere een. Det nemmeste og hurtigste er, uden tvivl, at gøre det første.**

**Spillet skal selvklart udstyres med en stærk beskyttelse. Piratkopier bliver spredt med lysets hastighed, når først en enkelt pirat har fået fingrene i et eksemplar. Der findes mange beskyttelsesprogrammer. Bl.a. et ganske udmærket, der hedder Underground Network. Det kan man bruge til, at lægge fejl ned på disketten med. Så kan man, når man loader sit spil, bare tjekke og se om fejlen er på disk'en, og samtidigt beskytter det mod kopiering. Meeen, for at opnå den ultimative beskyttelse, må man selv gribe tastaturet, og lave en beskyttelse selv. Det er ganske besværligt, men hvis man sælger på royalty-basis, vil det i den sidste ende nok vise sig, at være umagen værd.**

# IC RUN

**Tegn abonnement på IC RUN idag og du får helt gratis to fuldprisspil på bånd**

**Det drejer sig om et af verdens absolut bedste spil til Commodore 64: BOULDER DASH CONSTRUCTION SET, en anmelderfavorit, der har solgt i over 300.000 eksemplarer.**

**Ved siden af får du »Indoor SPORTS«, også et Commodore 64-spil, der er så godt at en AMIGA-version snart er på trapperne.**

★ For de seriøse med diskstation, kan der istedet vælges et praktisk tekst-behandlingsprogram, nemlig BWW Tekst 1 til Commodore 64,

★ AMIGA GOLD er IC RUN's nyeste tilbud til Amiga ejere. En diskette fyldt med nogle af de mest interessante Public Domain og ShareWare programmer på markedet. AMIGA GOLD indeholder over 25 programmer, lydeksempler og DPaint billeder, der virkelig viser hvad Amigaen duer til.


## kun kr. 285,-

### NU OGSÅ AMIGA Virus-killer!

Jeg ønsker istedet AMIGA virus killer -diskette

*Klip kuponen ud og send den i en lukket kuvert til*

**IC RUN** · Vejlebrovej 116 2635 Ishøj

 Jeg ønsker at tegne helårsabonnement på RUN, så jeg automatisk får bladet tilsendt med posten.  
Pris kr. 285,- incl. moms, for 10 blade (12 numre).

*Klip kuponen ud og send den i en lukket kuvert til*

**IC RUN** · Vejlebrovej 116 2635 Ishøj

Navn \_\_\_\_\_

Adresse \_\_\_\_\_

Postnr. \_\_\_\_\_ By \_\_\_\_\_

- og Jeg ønsker 2 spil tilsendt (bånd)  
 Jeg ønsker istedet diskette med BWW Tekst 1  bånd  
 Jeg ønsker istedet AMIGA-diskette med ca. 20 pgr.  
 istedet ønsker jeg Commodore 128 disketten med ca. 20 pgr.  
 Jeg ønsker istedet 11 blade kr. 285,-

Danmarks laveste abonnementspris!

HJEMMECOMPUTER

Nr. 1/2 Januar/Februar 1988

5. Årgang

Kun Dkr. 28,50

Nok. 28,50



# LRUN

ALT OM COMMODORE

*Nu kan din AMIGA køre C-64 programmer!*

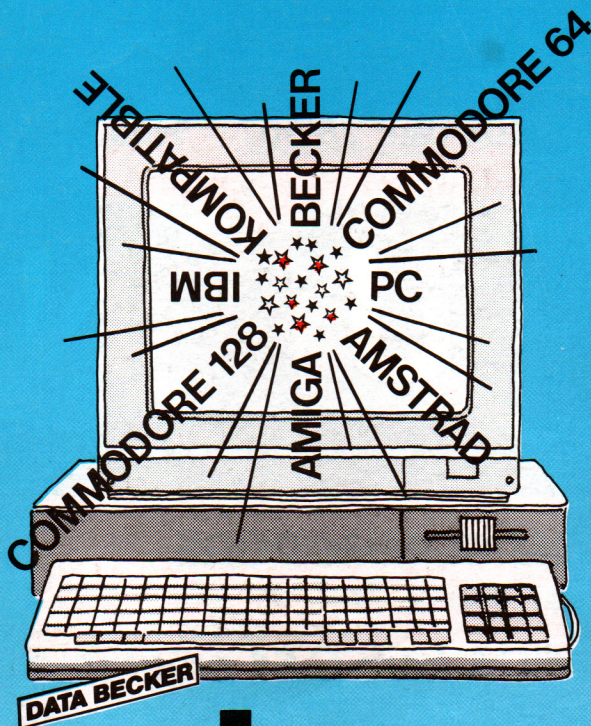
*Virus angreb på din Amiga?*

*Ny Commodore 64*

*AMIGA 3000 på vej?*



**Masser af programmer til  
Commodore 64 og 128**



# det bedste fra udlandet har vi oversat til dansk

## Commodore 128

### Førstebogen

En letfattelig indføring i C128 ..... 169,-

### Tips & Tricks

En guldgrube af format ..... 248,-

### Intern

Simpelthen BIBELEN ..... 348,-

### Peeks & Pokes

Få styr på de to kommandoer ..... 199,-

### Hardware-udvidelse

Byg selv dit tilbehør ..... 269,-

### CP/M øvelsesbog

Sådan fungerer CP/M 2.2 & 3.0 ..... 248,-

### Maskinsprog

Lynhurtige programmer med maskinsprog ..... 199,-

### Store floppybog

BIT-NUSSERENS DRØM ..... 298,-

## Commodore 64

### Store floppybog

Alt om 1541/70/71 med komplet kommenteret ROM-listning ..... 298,-

### Hardware-udvidelse

Byg selv dit tilbehør ..... 269,-

### CP/M Øvelse

CP/M systemet og alle kommandoer ..... 248,-

### Maskinsprog

Lynhurtige programmer i maskinsprog ..... 199,-

## Amstrad 464/664 & 6128

### Peeks & Pokes

Uundværlig for programmører ..... 199,-

### Tips & Tricks

Guldgrube for brugere af Amstrad ... 248,-

### Intern

Simpelthen bibelen ..... 348,-

### CP/M øvelsesbog

CP/M systemet & alle kommandoer ..... 248,-

### Første-bogen

En letfattelig indføring i Amstrad 664 & 6128 169,-

## Atari ST

### Tips & Tricks

Masser af rutiner til din Atari ST ... 269,-

## Amiga 500 - 1000 - 2000

### Tips & Tricks

En samling af høj karat i vejledninger af fremskreden programmering af »Pokes« og andre nyttige rutiner .... 248,-

### Maskinsprog

Du vil opdage, at maskinkodeprogrammering på Amiga er legende let ..... 248,-

### C for begyndere (med flere)

Det lynhurtige programmeringssprog C er som skabt til hurtige maskiner - Amiga ..... 248,-

## PC'ere IBM & kompatible

### PC begynderbog for viderekomne

DOS-grundkommandoer, bat-filer, edlin-editor, diskette & harddisk, installation m.v. .... 248,-

### PC Maskinsprog

Assemblerprogrammering, Debug, anvendelse af BIOS & DOS rutiner, adresserings-teknik og meget mere ..... 248,-

### C for begyndere (med flere)

Sådan lærer du det nye superhurtige programmeringssprog, hvor nøgleordene er hastighed, programmeringsteknik, biblioteker m.v. .... 248,-

### PC floppy & harddisk

Alt om organisation, reorganisering, sikkerhedskopier m.v. .... 348,-

Mange eksempler, inkl. diskette .... 348,-

Katalog ..... 5,-

## BECKERbase PC

Nu er det slut med de dyre løsninger på specialprogrammer, hvor man alligevel er låst fast på »standardrutiner«. BECKERbase PC kan arbejde med sammenhørende filer (relationsdatabase), d.v.s. at f.eks. 3 databaser som: kunde, produkt og faktura er opstillet i et hierarki: Hvorefter der kan søges:

Hvem har købt hvilke produkter?

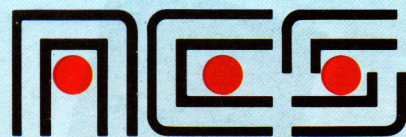
Hvilke fakturaer indeholder hvilket produkt? o.s.v.

Sådanne familiære forbindelser findes kun i databaser, hvor du frit kan definere en netværksstruktur.

Fra BECKERbase PC's hovedmenu kan du som programmør skabe ydedygtige brugerprogrammer og applikationer. Den kommende »bruger« vil ikke se BECKERbase PC, men kun de menuer og skærm-billeder, som er skabt af programmøren. IMPORT & EXPORT fra andre programmer gør BECKERbase til et virkeligt smidigt program, der kan bruges til stort set alle applikationer. Simpelthen markedets bedste til prisen .... Kun 998,-

## Familieprogram PC

(alle IBM kompatible) ..... 298,-  
samt C 64 & C 128 ..... 298,-



NORDIC COMPUTER SOFTWARE  
SMEDEGADE 7 · DK 6950 RINGKØBING

Kundeservice:  
Tlf. 0732 0202 kl 900-1000