

# Boulder-Dash-

**Das Spiel »Boulder-Dash« zieht heute noch viele in seinen Bann. Mit unserem Construction-Kit für den Schneider-CPC können Sie neue Level entwerfen, Objekte definieren, kurz: noch viel mehr Spaß haben.**

**A**ls wir vor fast vier Jahren das erste Boulder-Dash zum Testen bekamen, konnte keiner ahnen, daß dieses Spiel so ein Renner wird. Die Atari XL-Version erntete Kommentare wie »ganz nett« und wurde zur Seite gelegt. Erst Wochen später brach das Fieber aus und als die C 64-Version erschien, war keiner mehr zu bremsen: Durchspielte Nächte, Blasen an den Fingern, belagerte Computer. Und noch heute ist Boulder-Dash ein heißes Spiel, sei es als Version I, II, III, IV, als Remake oder als Construction-Kit wie eben unser »BDCK« für den Schneider-CPC.

Starten Sie als erstes die Programme »BDCK5.BAS« und »BDCK6.BAS«. Daraufhin werden die benötigten Maschinencode-Routinen erzeugt und auf Diskette abgelegt. »BDCK4.BAS« überträgt Boulder-Dash von Kassette auf Diskette. Legen Sie dazu die Original-Kassette ein und starten Sie das Programm.

Nach dem Start von »BDCK1.BAS« fragt Sie das Programm nach bereits vorhandenen Levels auf der Diskette. Falls Sie das erste Mal mit »BDCK« arbeiten haben Sie aber noch keine Level, und antworten mit »N«. BDCK erzeugt dann 20 leere Level. Falls Sie mit »J« geantwortet haben, geben Sie bitte den maximal vier Zeichen langen Namen der Leveler gestalten Level ein. Auf der Diskette werden die Level unter »FICT« plus ihrer eingegebenen vier Zeichen gespeichert. Wenn Sie nur <ENTER> drücken, werden die Level mit dem Namen »PICTURES« geladen. Nachdem die Level der aktuellen Arbeitsversion geladen oder erzeugt sind, wird der Hauptteil automatisch nachgeladen (»BDCK3.BAS«).

Sie können die aktuellen Level jederzeit speichern. Sie bestimmen dann, ob die Level im Speicher (»Memory«) oder auf Diskette (»Disk«) abgelegt werden sollen. In den Speicher paßt jedoch immer nur ein Level

## RAM-Disk für rasche Spiele

Mit »Drive« definieren Sie das augenblicklich anzusprechende Laufwerk. Zu Beginn ist das Programm auf »A« voreingestellt. Mit »M« wird eine eventuell vorhandene RAM-Disk angesprochen. Unterstützt wird die Vertex-RAM-Disk, die mit den Befehl »IMD« oder »IM« angesprochen wird. Besitzen Sie eine andere RAM-Disk, dann ersetzen Sie in Zeile 550 den Befehl »IM« durch den entsprechenden Befehl Ihrer RAM-Disk. Benutzen Sie die RAM-Disk aus Happy-Computer 10/87, ersetzen Sie in Zeile 550 »IM« durch »IRAM« und in Zeile 490 »IDLOGIN« durch »IDISC«.

Wenn Sie ein Bild ändern möchten, geben Sie einfach eine Zahl zwischen 1 und 20 ein. Die Zahl entspricht der Level-Nummer im Boulder-Dash. Nach der Eingabe befinden Sie sich im Level-Menü.

»A« ruft den Editor auf. Hier kann das Bild gelöscht, neu aufgebaut oder geändert werden.

<DEL> löscht die Zeichenfläche eines Bildes. Bei Boulder-Dash werden die einzelnen Bildelemente mit

ihren Koordinaten gespeichert. So sind ganze Linien oder Kästen möglich. Mit <DEL> wird die Zeichenfläche zwar gelöscht, das Bild bleibt aber erhalten. Der Zeiger, der auf die Adressen der einzelnen Teile zeigt, wird auf den Anfang des Bildes gesetzt.

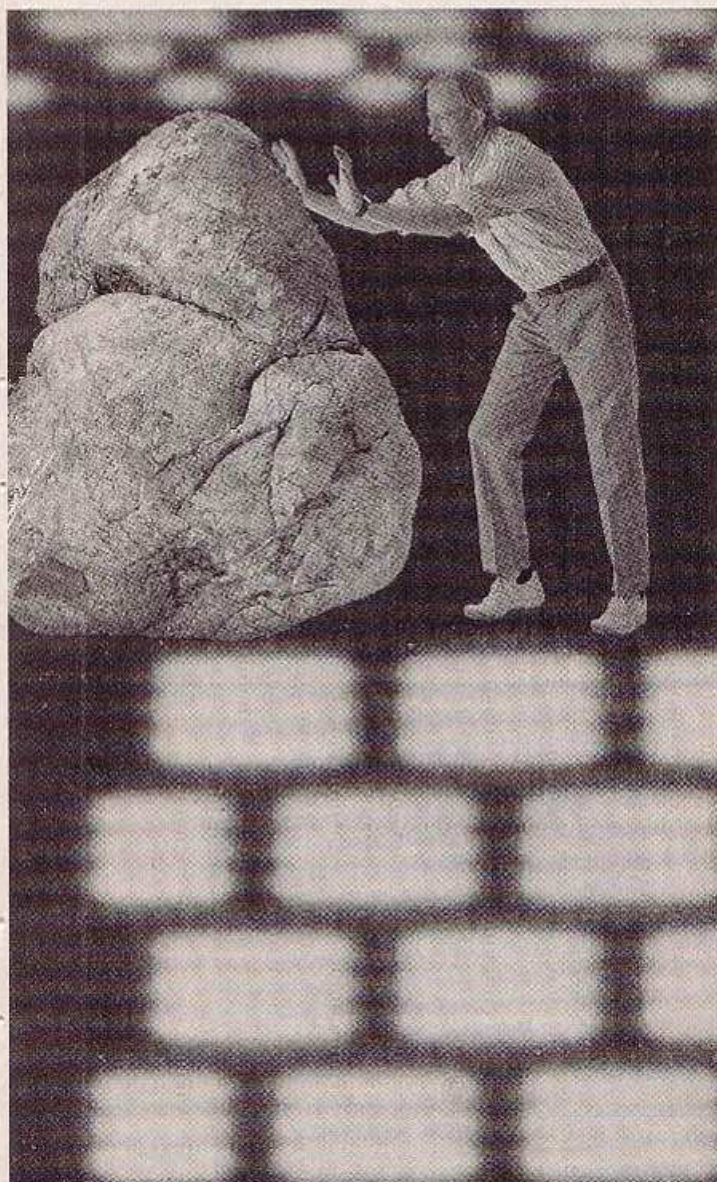
Mit <TAB> gehen Sie schrittweise vorwärts. <CLR> löscht den letzten Schritt. Das zuletzt erzeugte Teil, wird auf dem Bildschirm übermalt und aus dem Speicher entfernt, so daß Platz frei wird. Der freie Platz kann wiederum von anderen Teilen belegt werden. Mit <SHIFT+DEL> löschen Sie alles, was auf dem Bild zu sehen ist, auch im Speicher, wo wieder Platz freigegeben wird. Das Bild wird dann mit den noch verbleibenden Teilen neu aufgebaut. Mit <DEL> und anschließend dem Druck auf <SHIFT+DEL> gelangen Sie an das Bildende. Hier können Sie weitere Teile (Items) anhängen. Sie dürfen auch, während Sie mit <TAB> durch das Bild gehen, Teile einfügen. Bedenken Sie aber, daß nachfolgende Teile über Ihren Einfügungen liegen können. Sie dürfen so viele Teile einfügen und plazieren, bis der Speicher voll oder die Gesamtschrittzahl eines Bildes (117) überschritten ist. Beide Werte werden ständig am unteren Bildschirmrand angezeigt.

Mit den Cursortasten bringen Sie den Cursor an eine gewünschte Bildschirmposition. Dort setzen Sie mit den Buchstabentasten (auf Groß- und Kleinschreibung achten) und <SPACE> ein Item (siehe Tabelle).

Sie können aus diesen Teilen Linien und Kästen (auch gefüllte) bilden. Mit den Tasten 0 bis 4 wählen Sie den Aktionsmodus. Er wird ständig am unteren Bildrand angezeigt. Ein Einzelteil wird mit »0« oder mit »4« gewählt, eine Linie mit »1«. Sie kann horizontal, vertikal und diagonal sein. Sie bestimmen die Position, an der die Linie anfangen soll und drücken dann die Taste des Items, aus der die Linie bestehen soll. Mit der Cursortasten legen Sie die Richtung der Linie fest. Mit <SHIFT+ -> und <SHIFT+ -> ändern Sie die Länge. Wenn Sie den Endpunkt gefunden haben, dann drücken Sie eine beliebige Itemtaste. Die Linie wird danach gezeichnet.



# Construction-Kit



Wenn Sie bei einer Linie keinen Endpunkt haben, sondern zweimal die Itemtaste betätigen, wird die Länge und Richtung der alten Linie übernommen.

Ein Kasten ist ein Rechteck, das einen Rahmen aus einem Item und einen Inhalt aus einem anderen Item besitzt. Wählen Sie einen Kasten mit 2, einen Rahmen mit 3. Gehen Sie mit dem Cursor dahin, wo die linke obere Ecke sein soll. Dort betätigen Sie die Taste des Items, das den Rahmen bilden soll. Daraufhin können Sie mit den Cursortasten die Größe bestimmen. Jetzt legen Sie beim Kasten den Inhalt fest.

Wenn Sie das Bild zum ersten Mal eingegeben haben, müssen Sie die Parameter ändern. Dazu wählen Sie »P« im Untermenü. Nun werden die Parameter der Reihe nach abgefragt. In Klammern ist der alte Wert zu sehen. Falls er beibehalten werden soll, drücken Sie <ENTER>. Andernfalls geben Sie den neuen Wert ein. Da Boulder-Dash fünf Schwierigkeitsgrade anbietet, können Sie einige Werte für die Level getrennt eingeben. Hier sollten Sie steigende Schwierigkeitsgrade be-

rücksichtigen. Die Werte im einzelnen:

Rappelzeit (1-255): Mit diesem Wert bestimmen Sie, wie lange die Amöbe braucht, um sich in Steine zu verwandeln. Diese Zahl ist jedoch kein Zeitfaktor, denn der richtet sich auch nach der Größe der Amöbe. Gleichzeitig gib der Wert aber die Zeit an, die die magische Verwandlungsmaterie in Betrieb sein darf.

Punkte vorher (0-255): Die Punktzahl, die Sie pro Diamant erhalten, bis der Ausgang offen ist.

Punkte nachher (0-255): Für weitere Diamanten, die nach dem Öffnen des Ausganges gesammelt werden, können Sie eine andere Punktzahl festlegen.

Vorgabe Level 1-5 (1-255): Das Spiel kann zu den Items per Zufall positionierte Gegenstände verteilen.

Diamanten benötigt (1-99): Hier wird angegeben, wie viele Diamanten der Spieler sammeln muß, bis er den Ausgang benutzen darf.

Zeit (1-255 oder 0): Vergeben Sie die Zeit großzügig. Ein Wert von Null bedeutet: 999, und sollte nur in sehr schweren Level vergeben werden.

Zufallsgegenstand (0-63) und Anzahl Gegenstand (0-255): Bis zu vier Zufallsgegenstände können verteilt werden. Geben Sie die Nummer des Items ein und dann die Anzahl. Eine Anzahl von Null bedeutet kein Item. 255 bedeutet ein volles Bild. Wichtig: Die Startposition des ersten Items der Nummern 1 bis 4 wird immer neu geholt. Das heißt bei der Eingabe von »16«, »20«, »8«, »20« (16 ist ein Stein, 8 ein Wanderkästchen) und danach viermal der »0«, werden nicht genausoviel Steine wie Wanderkästchen verteilt. Stattdessen finden Sie nur Wanderkästchen, da diese auf die gleichen Positionen wie die Steine gelegt wurden. Sie müssen in diesem Beispiel die »20« durch eine »40« ersetzen.

Haben Sie alle Parameter eingegeben, befinden Sie sich wieder im Level-Untermenü. Von da gelangen Sie mit »H« ins Hauptmenü. Wenn alle Änderungen gemacht und die Level gespeichert sind, können Sie das Spiel mit »P« starten. Wenn Sie ein Spiel vom Editor aus starten, werden die Level genommen, die momentan im Speicher stehen. Wenn Sie das Spiel starten, muß es auf der gleichen Diskette wie der Editor sein.

(Ralph Hinnenberg/hf)

## BDCK ★★★

von **Andreas Rost**

Computertyp	CPC 464/664/6128
Sprache:	Basic und Maschinensprache
Eingabehilfe:	Explora/CP
Kurzbeschreibung:	Editor für Boulder-Dash
Länge auf Diskette	BDCK1.BAS: 2K (2048 Byte) BDCK2.BAS: 2K (2048 Byte) BDCK3.BAS: 9K (9216 Byte) BDCK4.BAS: 1K (1024 Byte) BDCK3.BIN: 1K (1024 Byte) BDCK3.BIN: 1K (1024 Byte)

- \* ist schnell abgetippt
- \*\* nehmen Sie sich etwas Zeit
- \*\*\* besser am Wochenende

## Funktionstasten beim Level-Editor

Taste	Bedeutung
<SPACE>	Zwischenraum
a	Außenmauer (unzerstörbar)
d	Diamant
e	Erde
f	Schmetterling
h	Bereits offener Kasten (Hutkasten), w.rld nur zu Testzwecken benötigt
i	Rockfort (im Starkasten); auf diesem Teil stellt sich die Bildschirm-Scrollposition ein
k	Ausgang: Kasten, der sich nach festgelegter Diamantenanzahl öffnet
m	Steinmauer
r	Amöbe (Rappelteil)
s	Stein
u	Unsichtbares Hindernis
v	Magische Verwandlungsmauer
w	Wanderkästchen (Firefly)
z	Zerstörbare Mauer, die wie die Außenmauer aussieht
E	Explosion, die zum Leerraum wird
Y	Explosion, die zum Diamanten wird

```

10 MODE 2:SYMBOL AFTER 256:MEMORY 38999:
SYMBOL AFTER 128:F=HMEM+1:OPENOUT"d"
:MEMORY 19999:CLOSEOUT:LOAD"bdck6.bin"
:CALL 39000 [A986]
130 INPUT"Version laden?",a$:IF a$="j"OR
a$="n"THEN GOSUB 350 ELSE IF a$="n"
HEN GOSUB 210 ELSE 130 [1468]
140 !LDIR,20000,21560,1559:LOAD"hdck5.bi
n",F:GOSUB 420:MODE 2:CHAIN"hdck3.ba
s" [5222]
210 S=20001:POKE 20000,5:WINDOW 10,20,1,
1:FOR i=1 TO 20:LOCATE#1,1,1:PRINT#1
,i:RESTORE 350:POKE S,i:S=S+1:FOR j=
1 TO 3:PRINT j:READ p:POKE S,p:S=S+1
:NEXT j:FOR j=4 TO 8:PRINT j:p=INT(R
ND*254+1):POKE S,p:S=S+1:NEXT j:FOR
j=9 TO 22:PRINT j:READ p:POKE S,p
280 S=S+1:NEXT j:p=INT(RND*50+5):FOR j=2
3 TO 26:PRINT j:POKE S,p:S=S+1:p=p-I
NT(RND*10+1):IF p<0 THEN p=0 [241K]
310 NEXT j:FOR j=27 TO 33:PRINT j:READ p
:POKE S,p:S=S+1:NEXT j:NEXT i:RETURN [22D0]
350 DATA255,10,15 [0ABC]
360 DATA10,11,12,13,14,250,240,230,215,2
00,16,20,2,8 [B31A]
370 DATA37,4,4,4,50,10,255 [3A3A]
390 INPUT"Version",a$:IF a$="n"THEN a$="u
res" [69A8]
400 LOAD"picot"+a$,20000:RETURN [6A72]
420 KEY DEF 72,1,240:KEY DEF 73,1,241:KE
Y DEF 74,1,242:KEY DEF 75,1,243:KEY
DEF 76,1,13:KEY DEF 77,1,42:KEY DEF
9,0,13:KEY DEF 79,1,127,126,1:KEY DE
F 68,1,124:SYMBOL 211,102,0,120,12,1
24,204,118,0:SYMBOL 212,102,0,60,102
,102,102,60,0 [D4G0]
530 SYMBOL 213,102,0,102,102,102,124
,0:SYMBOL 214,102,0,60,102,102,126,1
02,0:SYMBOL 215,198,0,124,198,198,19
8,124,0:SYMBOL 216,198,0,198,198,198
,198,124,0:RETURN:SAVE"m:b-edit":ON
ERROR GOTO 520:MD:RUN [EFF4]
620 RESUME 650 [88FE]
630 RESUME 640 [9388]
640 :C:RUN [4E88]
650 ON ERROR GOTO 630:MD:RUN [4450]

```

Mit »BDCK1.BAS« starten Sie das Construction-Kit

```

10 IF S0=0 THEN S0=20000:MEMORY S0-1:INP
UT"Version",V$:IF V$="n"THEN LOAD"picot
ures",S0 ELSE LOAD"picot",4V$ S0 [5708]
20 IF V$="j"OR V$="ha"THEN V$="dev" [EACE]
30 IF PEEK(39000)<33 THEN LOAD"mcprint3
":CALL 39000 [19F2]
40 MCDE 1:po=S0+1560:;BILD,20,S0,1559,1:

```

»BDCK2.BAS« wird nachgeladen

```

O-PEEK(330)+256*PEEK(331)-1-S0:WINDOW
1,40,1,23:m=20:FOR i=1 TO m:PAPER 0:
PEN 1:;BILD,i,S0,1559,1:S1=PEEK(330)+
256*PEEK(331) 1:E=PEEK(332)+256*PEEK(
333)-1 [AD24]
90 IF S1=65534 THEN PRINT i:"Gnicht gefu
nder":IF i<17 THEN 130 ELSE 1=0:GOTO
120 [FBD8]
100 1=S1-S0:LOCATE#1,1,24:PRINT#1,"Offse
t"i="-1 [A718]
120 POKE po,1 MOD 256:POKE po+1,1\256 [3218]
130 po=po+2:NEXT:;LDIR,S0,49152,1600:MEM
ORY 7315:LOAD"bmc2":RESTORE:FOR i=8
180 TO 819:READ p:POKE i,p:NEXT:DAT
A&21,0,&C0,&11,&f7,&7e,1,&17,&06,&ed
,&10,&c9 [57A2]
260 GOSUB 330:CALL 8180:IF PEEK(49152)<>
5 THEN GOSUB 400 [29D6]
290 LOCATE 1,20:PRINT"Poke (y/n)?:FOR i
=150 TO 1 STEP-1:LOCATE 22,20:PRINT
i:1$=INKEY$:IF 1$=""THEN NEXT i [571A]
305 IF 1$="s"THEN SAVE"bmc2",b,&2000,&7
FC0,&9025 [D70C]
310 IF 1$<>"n"THEN POKE &3981,190 [95FC]
320 CALL &9025 [3CCE]
330 so=49152+1560:D0=&50D0+&1E00:FOR i=1
TO 20:POKE D0,PEEK(so):POKE D0+1,PE
EK(so+1):so=so+2:D0=D0+2:NEXT:RETURN [DFC2]
400 LOAD"exar"+V$,7316:FOR n=0 TO 15:I=2
4512+(n\4)*&1E0+(n MOD 4)*54:S=7316+
n*54:FOR i=0 TO 53:POKE D+i,PEEK(S+i
):NEXT i:NEXT n:POKE 32503,5:RETURN:
SAVE"m:offset":CHAIN"a:bdck3.bas",34
60 [7GG2]

```

»BDCK2.BAS« (Schluß)

```

10 CLEAR:;RILD,20,20000,1559,1:en=PEEK(3
32)+256*PEEK(333):IF en=65535 THEN IN
K 1,26:ERROR 4 ELSE F=21559-en [33B8]
30 DEFINTx,y:GOSUB 3240:1$="a":GOSUB 430
:B$="<G240>":R1$="<G241>":L3$="<G242>
":R0$="<G243>":S0$="<G248>":S1$="<G24
7>":S2$="<G244>":S3$="<G245>":D0$="<G2
40><G241><G242><G243>":E$=CHR$(13):D$
=CHR$(127):C1$=CHR$(16):C$=CHR$(20):C
0$=CHR$(19):N$=CHR$(1) [59D0]
120 a=REMAIN(0):CALL &BB7E:MODE 1:nr$=""
:GOSUB 3150:PRINT"1-20 ... Bild:PRI
NT"<3>P ... Spielen":PRINT<3>S ...
Speichern":PRINT<3>L ... Laden":PRI
NT"<3>C ... Catalog":PRINT<3>D ...
Drive .. Aktuell:UPPER$(dr$):LOCATE
#1,1,7:PRINT#1,C$;Wah:" [2F6C]
220 EVERY 20 GOSUB 730:CALL &BB7B:EI:x=6 [E142]
240 LOCATE x,7:1$=LOWER$(INKEY$):IF 1$>
"0"AND 1$<="9"THEN LOCATE#1,x,7:PRIN
T#1,1$:x=x+1 [FA54]
270 IF(1$="3"AND 1$<="9")OR 1$="0"THEN
nr=VAL(nr$+1$):GOTO 810 [1142]
280 IF 1$="1"OR 1$="2"THEN IF nr$=""THEN
nr$=1$ELSE nr=VAL(nr$+1$):GOTO 810 [E7AE]
290 IF 1$=E$OR 1$=" "AND nr$<>""THEN nr=
VAL(nr$):GOTO 810 [482E]
300 IF 1$="s"THEN GOSUB 3130:GOTO 120 [2984]
310 IF 1$="l"THEN GOSUB 2610:GOTO 120 [A67C]
320 IF 1$="p"THEN S3=20000:CHAIN"offset"
[79A2]
330 IF 1$="c"THEN 360 [E482]
340 IF 1$="d"THEN GOSUB 440:GOTO 120 [8B10]
350 GOTO 240 [F84E]
360 MCDE 2:DI:CALL &BB7E:CAT:CALL &EB18:
CALL &BB7B:EI:GOTO 120 [5C6A]
440 DI:CALL &BB7E:1=3:r=11:o=5:u=7:GOSUB
3330:WINDOW SWAP 4,0:PRINT"W<G211>h
le ... A<3>B<3>M": [5446]
460 1$=INKEY$:IF 1$=""THEN 450 [9A1C]
470 I=INSTR("abm",1$):IF I=0 THEN 460 [4080]
480 ON ERROR GOTO 630:IF 1$>="a"AND 1$<="
d"THEN;DLOGIN,ASC(1$)-97 [6D76]
500 ON ERROR GOTO 640:IF 1$="a"THEN;A:GO
TO 680 [310E]
520 IF 1$="b"THEN;B:GOTO 680 [B5EE]
530 ON ERROR GOTO 650:IF 1$="m"THEN;MD:G

```

»BDCK3.BAS« ist das eigenliche Construction-Kit



# Schneider Spiele-Listing

<pre> x=40 2400 IF l\$=R0\$THEN x-x+1:IF x&gt;40 THEN x= xo+M [3CD6] 2410 IF l\$=R1\$THEN y=y+1:IF y&gt;22 THEN y= yo+M [C4CC] 2420 IF l\$=B\$THEN y=y-1:IF y&lt;yo+M THEN y =22 [99DA] 2430 GOSUB 2210:LOCATE x,y:GOTO 2350 [7066] 2450 IF A1=2 AND l\$=E\$THEN 2330 [9DP4] 2460 a(0)=a(0)+128:IF A1=3 THEN a(0)=a(0) )+64 [40E8] 2470 a(5)=INSTR(T\$,l\$)-1:a(3)=x-xo+1:a(4) )=y-yo+1:GOTO 1720 [C826] 2500 x=xo:y=yo:GOTO 1230 [C6E8] 2550 BORDER 26:INK 0,26:INK 1,26:INK 2,2 6:INK 3,26: SOUND 2,50,7,7:WHILE BQ( 2)&lt;&gt;4:WEND:GOSUB 3240:RETURN [19E2] 2610 f1\$="1" [8C98] 2860 DI:CALL &amp;BB7E:CALL &amp;BB81:DI:1=2:r=1 5:o=2:u=3:GOSUB 3330:WINDOW SWAP 4, 0:PRINT"M ... Memory":PRINT"D ... D isk":LOCATE 1,2 [3244] 2680 l\$=INKEY\$:IF l\$="m"THEN 2780 [09D2] 2700 IF l\$="d"OR l\$=E\$THEN 2720 [D13C] 2710 GOTO 2680 [2526] 2720 CALL &amp;BB73:DI:CALL &amp;BB81:1=3:r=20:c =4:u=4:GOSUB 3330:WINDOW SWAP 4,0:C ALL &amp;BB7B:INPUT"Version";V\$:IF f1\$= "1"THEN IF V\$="THEN LOAD"pictures" ,20000 ELSE LOAD"pict"+V\$,20000 ELG E IF V\$="THEN SAVE"pictures",b,2000 00,1559 ELSE SAVE"pict"+V\$,b,20000, 1559 [7598] 2750 CALL &amp;BB7B:EI:BILD,20,20000,1559,1 :en=PEEK(332)+256*PEEK(353):IF en=6 5535 THEN INK 1,26:ERROR 4 ELSE F=2 1559-en [B112] 2770 RETURN [8CA2] 2780 IF f1\$="1"THEN LDIR,21560,20000,155 9 ELSE LDIR,20000,21560,1559 [3FB2] 2790 GOTO 2750 [6A32] 2810 a\$=CHR\$(p+128):RETURN [509E] 2860 A0=s-1:PRINT"Bild"PEEK(st) ... Pa rameter:":PRINT"Rappelzeit("PEE K(A0+2):INPUT"):p\$:IF p\$&lt;&gt;"THEN POKE A0+2,VAL(p\$) [DAA0] 2890 PRINT"Punkte, vorher("PEEK(A0+3):IN PUT"):p\$:IF p\$&lt;&gt;"THEN POKE A0+3,V AL(p\$) [9F8E] 2900 PRINT"Punkte, nachher("PEEK(A0+4):I NFUT"):p\$:IF p\$&lt;&gt;"THEN POKE A0+4, VAL(p\$) [3D03] 2910 FOR i=1 TO 5:PRINT"Vorgabe Level"i" ("PEEK(A0+4+i)"):INPUT p\$:IF p\$&lt;&gt; </pre>	<pre> ""THEN POKE A0+4+i,VAL(p\$) [5BA2] 2930 NEXT:FOR i=1 TO 5:PRINT"Diananten b enoetigt Level"i("PEEK(A0+9+i)"): :INPUT p\$:IF p\$&lt;&gt;"THEN POKE i+A0+9 ,VAL(p\$) [7604] 2950 NEXT:FOR i=1 TO 5:PRINT"Zeit fuer L evel"i("PEEK(A0+14+i)"):INPUT p\$ :IF p\$&lt;&gt;"THEN POKE A0+i+14,VAL(p\$) [EFB0] 2970 NEXT:CALL &amp;BB7E:GOSUB 3060:CALL &amp;BB 7B:FOR i=1 TO 4:PRINT"Zufallsgogens tand"i("PEEK(A0+19+i)"):INPJT p\$ :IF p\$&lt;&gt;"THEN POKE A0-i+19,VAL(p\$) [5F6C] 3030 PRINT"Anzahl Gegenstand"i("PEEK(A0 +23+i)"):INPUT p\$:IF p\$&lt;&gt;"THEN P OKE A0+i+23,VAL(p\$) [913A] 3040 NEXT:GOTO 810 [20A4] 3060 CLS:FOR i=0 TO 3:i=1*10+2:r=i*10+9: o=2:u=17:GOSUB 3330:WINDOW SWAP 4,0 :FOR j=0 TO 15:p=1*16+j:GOSUB 2810: PRINT USING"###";p:PRINT" = "a\$:NE XT j,i:WINDOW 1,40,20,25:RETURN [AAD0] 3130 f1\$="s":GOTO 2660 [E664] 3150 PEN 1:PAPER 0:PEN#1,1:PAPER#1,3:WIN DOW#2,1,40,23,25:WINDOW#3,2,39,2,21 :PAPER#2,0:PEN#2,1:PAPER#3,3:WINDOW #5,2,39,2,21:PEN#5,2:PAPER#5,3:TAG# 7:PRINT"WA":RETURN [433E] 3240 IF i0\$=""THEN PRINT"Farbe":PR INT"G ... Gruen":WHILE i0\$=""i0\$=I NKEY\$:WEND [EFA6] 3250 SPEED INK 20,20:IF i0\$="f"THEN i0=1 5:INK 0,13: BORDER 13:INK 1,0:INK 2, 1:INK 3,9 ELSE IF i0\$="g"THEN i0=26 :INK 0,0: BORDER 0:INK 1,26:INK 2,18 :INK 3,8 ELSE 3240 [C390] 3260 RETURN [8796] 3330 PAPER#4,0:PEN#4,1:WINDOW#4,1-1,r+1, o-1,u+1:CLS#4:WINDOW#4,1,r,o,u:11=1 *16-16:rr=r*16:oo=414-o*16:uu=398-u *16:PRINT"WD":PLOT 11-6,uu-4:DRAW rr+4,uu-4:DRAW rr+4,oo+4:DRAW 11-3, oo+4:DRAW 11-6,uu-4:PLOT 11-12,uu-1 0 [DC80] 3380 DRAW rr+10,uu-10:DRAW rr+10,oo+10:D RAW 11-12,oo+10:DRAW 11-12,uu-10:RE TURN [F524] 3410 1=20:r=39:o=24:u=24:GOSUB 3330:PRIN T#4,"Error":ERR;" in":END:KEY DEF 9,1,224:END:SAVE"m:beditp2.bcn" :CHAIN"a:b-edit",580 [1236] </pre>
---	---

»BDCK3.BAS« (Schluß)

<pre> 10 !TAPE. IN: !DISC. OUT: OPENOUT "d": MEMORY &amp;1FFF: CLOSEOUT: LOAD "loader", &amp;A000: POK E &amp;A014, 201: POKE &amp;A00E, 32: CALL &amp;A009: FOR i=&amp;9021 TO &amp;903D: READ p: POKE i, p: NEXT: DATA &amp;21, &amp;00, &amp;20 [F50C] 100 DATA &amp;11, &amp;00, &amp;02 [AA72] 110 DATA &amp;01, &amp;25, &amp;70 [978A] 120 DATA &amp;d, &amp;b0 [2E40] 125 DATA &amp;0, &amp;0, &amp;0, &amp;0, &amp;0, &amp;0, &amp;0, &amp;0 [5634] 130 DATA &amp;c3, &amp;52, &amp;1f [1258] 140 SAVE "bomc", b, 24823+&amp;1E00, 1559 [527C] </pre>	<p>»BDCK4.BAS« überträgt Boulder-Dash auf Diskette</p>
--	--

<pre> 100 ***** [A480] 110 * BDCK5.BAS - DATA-lader von 'CPC' * [BA1A] 120 ***** [1E84] 130 ***** [DFB6] 140 DATA A1C6,00,00,00,00,00,00,00,00,0000 [B04C] 150 DATA A1CE,08,22,90,42,04,11,44,10,0357 [CCDF] 160 DATA A1DE,FF,92,FF,44,FF,22,FF,34,1400 [C302] 170 DATA A1DF,FF,92,FF,55,00,FF,49,FF,1324 [892A] 180 DATA A1EG,FF,99,99,FF,FF,99,99,FF,1632 [0260] 190 DATA A1EE,FF,FF,C3,C3,C3,C3,FF,FF,1800 [8E9A] 200 DATA A1FG,FF,A5,A5,A5,FF,A5,A5,FF,1590 [324E] 210 DATA A1FE,FF,A5,FF,A5,A5,FF,A5,FF,1630 [119A] 220 DATA A205,FF,81,BD,A5,A5,BD,81,FF,1476 [610E] 230 DATA A20E,FF,89,99,BF,BF,99,89,FF,1472 [8B3E] 240 DATA A21F,FF,81,99,BD,FF,99,99,FF,1542 [5718] 250 DATA A21E,FF,91,99,FD,FD,99,91,FE,1612 [A128] 260 DATA A225,42,24,18,3C,18,3C,24,86,0408 [9B02] </pre>	<p>»BDCK5.BAS« erzeugt »BDCK5.BIN«</p>
---	--

<pre> 270 DATA A22E,AA,55,AA,55,AA,55,AA,55,1020 [99B8] 280 DATA A236,00,18,2C,42,2C,18,00,00,0202 [88C4] 290 DATA A23E,00,18,34,42,34,18,00,00,0218 [A43A] 300 DATA A246,3C,76,EB,F5,DF,FB,7A,3C,1314 [0808] 310 DATA A24E,DD,FF,62,5F,E7,FA,C7,FB,1600 [C052] 320 DATA A256,3C,7E,E7,E7,A5,C3,66,3C,1170 [5134] 330 DATA A25E,DD,FF,66,67,A5,C2,E7,FB,1522 [361E] 340 DATA A266,08,14,3E,41,3E,14,08,00,0245 [A6EC] 350 DATA A26E,18,3C,7E,DB,E7,DB,7E,3C,1065 [6B1A] 360 DATA A276,08,14,36,55,22,14,08,00,0229 [2BFA] 370 DATA A27E,18,3C,6E,E7,A5,C3,66,3C,0939 [BAB2] 380 DATA A286,FF,81,8D,91,89,B1,81,FF,1568 [94D0] 390 DATA A28E,00,00,00,00,FF,00,00,00,0255 [01D4] 400 DATA A296,18,3C,42,E7,E7,E7,66,3C,1005 [E07A] 410 DATA A29E,00,00,24,03,28,00,00,00,0084 [7DA0] 420 DATA A2A6,30,12,24,0A,2C,12,00,00,0134 [BED0] 430 DATA A2AE,42,89,24,59,3C,0A,91,44,0611 [1644] 440 DATA A2B6,00,08,24,58,24,0A,10,00,0194 [37D4] 450 DATA A2BE,30,00,14,03,14,08,30,00,0056 [10BC] 460 DATA A2C6,08,10,0A,41,0A,10,08,00,0133 [C6D8] 470 DATA A2CE,08,10,2A,41,0A,14,08,00,0169 [EB6] 480 DATA A2D6,08,14,2A,41,2A,14,08,00,0205 [B5F6] 490 DATA A2DE,08,14,2A,55,2A,14,08,00,0225 [1724] 500 DATA A2E6,08,14,3A,41,2E,14,08,00,0225 [27F8] 510 DATA A2EE,37,C7,EE,83,6F,D7,D9,FF,1580 [4060] 520 DATA A2FE,99,38,52,7C,94,29,A6,51,0851 [2C5A] 530 DATA A2FE,99,38,52,7C,94,29,A6,11,0782 [CD5E] 540 DATA A306,19,38,52,7C,90,28,26,11,0526 [4DF2] 550 DATA A30E,22,77,A8,27,22,AB,71,27,0717 [5648] 560 DATA A316,08,1C,2A,7F,2A,1C,08,00,0283 [4D48] 570 DATA A31E,42,24,7E,A5,FF,42,42,E7,1011 [2582] </pre>
---

```

580 DATA A328,81,C7,7E,3C,3C,7C,E3,81,1054 [F29E]
590 DATA A32E,00,18,3C,5E,34,18,00,00,0254 [0006]
600 DATA A338,24,42,FF,42,24,81,FF,44,0911 [444A]
610 DATA A33E,ED,00,BD,00,EF,00,DD,00,0886 [BBC2]
620 DATA A346,81,C3,A7,FF,FF,A7,C3,81,1492 [A1E4]
630 DATA A34E,FF,5A,3C,18,18,3C,7E,FF,0894 [8012]
640 DATA A356,81,C3,E5,FF,FF,E5,C3,81,1618 [39EE]
650 DATA A35E,FF,7E,3C,18,18,3C,5A,FF,0894 [7418]
660 DATA A366,E2,8F,E2,40,F8,22,EF,42,1246 [FAA4]
670 DATA A36E,E2,8F,02,40,F0,42,0F,E2,0982 [9A92]
680 DATA 9A08,7E,04,87,06,00,4F,09,56,0445 [2B2C]
690 DATA 9A10,23,5E,E1,DD,46,03,CD,DA,1071 [89BE]
700 DATA 9A18,9A,CD,35,9A,7C,82,87,7D,1048 [40B8]
710 DATA 9A20,83,8F,10,F2,01,05,00,3A,0564 [2A06]
720 DATA 9A26,D4,9A,FE,01,C8,FE,06,C8,1281 [EAE8]
730 DATA 9A30,DD,09,C3,41,99,3A,56,01,0788 [9A84]
740 DATA 9A36,DD,E5,CD,E5,9A,DD,E1,C9,1685 [5862]
750 DATA 9A40,F5,CD,09,BB,FE,71,28,40,1117 [FAB4]
760 DATA 9A48,F1,E6,3F,D5,16,01,FE,01,1025 [4792]
770 DATA 9A50,28,08,FE,07,28,22,FE,2B,0680 [9084]
780 DATA 9A58,20,04,16,03,18,1A,FE,3A,0423 [F43E]
790 DATA 9A60,28,14,FE,03,28,10,FE,06,0633 [3556]
800 DATA 9A68,38,0E,FE,33,30,0A,FE,0B,0698 [1ABE]
810 DATA 9A70,38,04,FE,2F,38,02,16,02,0443 [ED26]
820 DATA 9A78,F5,7A,32,54,01,CD,9D,9A,1018 [EAE6]
830 DATA 9A80,F1,E6,3F,32,56,01,D1,C9,1081 [0F74]
840 DATA 9A88,ED,7B,C2,9A,C9,00,FF,01,1165 [4EE6]
850 DATA 9A90,FF,01,00,01,01,00,01,FF,0514 [EF18]
860 DATA 9A98,01,FF,00,FF,FF,DD,E5,E5,1445 [1E3C]
870 DATA 9AA0,C5,CD,90,BB,C1,E1,DD,E1,1597 [1634]
880 DATA 9AA8,C9,21,FF,FF,22,4A,01,22,0887 [5RCA]
890 DATA 9AB0,4C,01,C9,21,C4,9A,7E,FE,1041 [72D0]
900 DATA 9AB8,FF,CA,88,9A,CD,E5,9A,23,1370 [2E30]
910 DATA 9AC0,18,F4,80,B0,50,61,72,61,0960 [EA2A]
920 DATA 9AC8,6D,65,74,65,72,20,45,72,0756 [C440]
930 DATA 9AD0,72,6F,72,FF,03,01,01,01,0600 [F72C]
940 DATA 9AD8,01,20,22,D7,9A,E5,24,2C,0745 [5774]
950 DATA 9AE0,CD,75,BB,E1,C9,32,D9,9A,1356 [D10E]
960 DATA 9AE8,F6,00,CD,5A,BB,C9,87,00,1192 [8EF2]
970 DATA 9AF0,02,28,16,01,FE,00,23,10,0375 [3A1E]
980 DATA 9AF8,FE,02,C2,B3,9A,DD,65,02,1108 [15E0]
990 DATA 9B00,DD,6E,00,22,D5,9A,3E,05,0799 [1FA6]
1000 DATA 9B08,3C,32,D4,9A,DD,21,EE,9A,1122 [1418]
1010 DATA 9B10,C3,41,99,00,00,00,00,00,0413 [30F2]
1020 DATA *END* [EA1A]
1030 adr=&9858:zeile=1040:MEMORY adr=1 [ECF2]
1040 READ d$:IF d$="*ENDE*"THEN 1140 [2A40]
1050 pr=0 [868A]
1060 FOR i=1 TO 8 [30C0]
1070 READ a$:a=VAL("&"a$) [AE9E]
1080 POKE adr,a:adr=adr+1 [GG7A]
1090 pr=pr+a [A2EE]
1100 NEXT i [A254]
1110 READ pr2 [A766]
1120 IF pr<>pr2 THEN PRINT"Pruefsummenfehler in Zeile";zeile:STOP [D288]
1130 zeile=zeile+10:GOTO 1040 [4270]
1140 SAVE"BDCK6.BIN",B,&9858,&2BC [17A2]
1150 PRINT d$:END [8D5A]

```

»BDCK6.BAS« (Schluß)

```

100 ***** [A480]
110 * BDCK6.BAS - DATA-Loader von 'CPC' * [5B1C]
120 ***** [1E84]
130 ***** [DF06]
140 DATA 9858,21,82,98,01,88,98,CD,D1,0952 [9F3E]
150 DATA 9860,EC,C9,00,00,00,00,77,98,0660 [25FE]
160 DATA 9868,C3,8F,98,C3,A7,98,C3,D3,1410 [F4A6]
170 DATA 9870,58,C3,BF,98,C3,F4,9A,4C,1359 [A7D2]
180 DATA 9878,44,49,D2,4C,44,44,D2,42,0839 [8142]
190 DATA 9880,49,4C,C4,49,54,45,CD,53,0859 [EE70]
200 DATA 9888,43,52,49,4E,49,D4,00,FE,0839 [0C62]
210 DATA 9890,03,C0,DD,4E,00,DD,46,01,0786 [F964]
220 DATA 9898,DD,5E,02,DD,56,03,DD,5E,0958 [DCDE]
230 DATA 98A0,04,DD,66,05,ED,B0,C9,FE,1200 [E0BE]
240 DATA 98A8,03,C0,DD,4E,00,DD,46,01,0786 [1C8A]
250 DATA 98B0,DD,5E,02,DD,56,03,DD,5E,0958 [72E6]
260 DATA 98B8,04,DD,66,05,ED,B8,C9,FE,1208 [7EFG]
270 DATA 98C0,01,C2,B3,9A,32,D4,9A,DD,1165 [E5A8]
280 DATA 98C8,66,01,DD,6E,00,E5,DD,E1,1109 [DCBE]
290 DATA 98D0,C3,41,99,ED,73,C2,9A,FE,1367 [48E2]
300 DATA 98D8,04,20,06,DD,23,DD,23,18,0570 [8F52]
310 DATA 98E0,05,FE,03,C2,B3,9A,32,D4,1051 [2176]

```

»BDCK6.BAS« erzeugt »BDCK6.BIN«

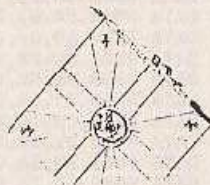
\*\*\*\*\*

## Postspiel

# H I S T O R I E

> 1 8 0 0 <

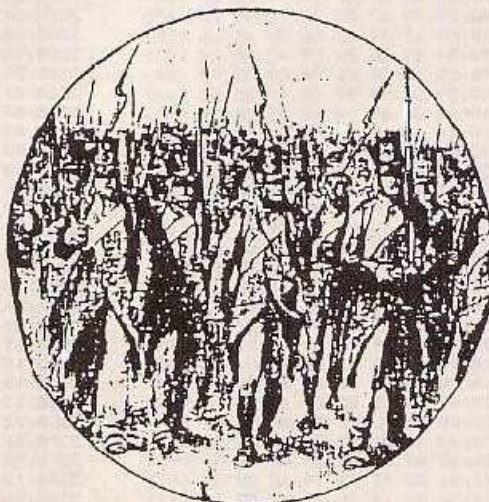
## KAMPF UM DIE WELTVORHERRSCHAFT



Werden Sie Herrscher eines Reiches in Europa zur Zeit der FRANZÖSISCHEN REVOLUTION!

Diplomatie, Politik, Strategie Kolonialismus, Welthandel, Wirtschaftsausbau und Geschick auf dem Schlachtfeld bestimmen den Weg Ihres Landes in einer packenden Simulation von DECOS.

Ihre Ideen; die von Spielern aus Deutschland und Europa gestalten den Verlauf eines von einem Computer verwalteten Briefspiels.



Nähere Informationen zu diesen und anderen anspruchsvollen BRIEFSIMULATIONSSPIELEN erhalten Sie von

**DECOS GmbH**  
Egenolffstr. 29  
6000 Frankfurt 1  
Tel. 069-499523

Achtung! Kein Software-Verkauf!

\*\*\*\*\*

# Schneider Spiele-Listing

```
320 DATA 98E8,9A,DD,4E,00,DD,46,01,DD,0966 [7BE8]
330 DATA 98F0,66,03,DD,6E,02,DD,56,04,0749 [3334]
340 DATA 98F8,23,7A,FE,15,D0,FE,01,D8,1111 [8734]
350 DATA 9900,28,22,D5,11,1B,00,19,D1,0565 [D476]
360 DATA 9908,3E,FF,BE,28,05,5D,B1,E2,1192 [BA36]
370 DATA 9910,A9,9A,7A,ED,A1,E2,A9,9A,1392 [1FE8]
380 DATA 9918,28,09,2B,3E,FF,BE,23,28,0674 [1D9C]
390 DATA 9920,F1,18,DF,2B,E5,DD,E1,11,1223 [5FAC]
400 DATA 9928,1B,00,19,3E,FF,ED,B1,E2,1009 [54A8]
410 DATA 9930,A9,9A,DD,22,4A,01,22,4C,0763 [FA62]
420 DATA 9930,01,DD,19,3A,D4,9A,FE,04,0929 [B9AA]
430 DATA 9940,C8,DD,7E,00,FE,FF,C8,DD,1477 [E43E]
440 DATA 9948,66,01,DD,6E,02,2D,2D,22,0560 [7652]
450 DATA 9950,58,01,CE,7F,20,14,CD,77,0793 [5000]
460 DATA 9958,C2,FE,99,C3,40,9A,CD,DA,1447 [4E24]
470 DATA 9960,9A,CD,35,9A,01,03,00,C3,0765 [1F50]
480 DATA 9968,27,9A,CE,77,F5,DD,7E,05,1112 [2DC2]
490 DATA 9970,CD,40,9A,32,57,01,3A,54,0703 [8B34]
500 DATA 9978,01,32,55,01,ED,7E,00,CD,0689 [BE5E]
510 DATA 9980,40,9A,CE,DA,9A,DD,46,03,1089 [F5C2]
520 DATA 9988,CD,35,9A,10,FB,2A,58,01,0810 [DC76]
530 DATA 9990,DD,7E,04,3D,85,6F,CD,DA,1079 [52FC]
540 DATA 9998,9A,DD,46,03,CD,35,9A,10,0876 [5FA0]
550 DATA 99A0,FB,DD,7E,04,D6,02,28,49,0531 [8EA6]
560 DATA 99A8,38,47,4F,2A,58,01,2C,CD,0586 [3898]
570 DATA 99B0,DA,9A,E5,CD,35,9A,DD,7E,1360 [AE2C]
580 DATA 99B8,33,84,3D,67,CD,DA,9A,CD,1081 [67F2]
590 DATA 99C0,35,9A,E1,F1,F5,20,27,E5,1218 [6A88]
600 DATA 99C8,DD,46,03,78,FE,03,38,14,0747 [387A]
610 DATA 99D0,3A,55,01,CD,9D,9A,05,24,0701 [9066]
620 DATA 99D8,05,CD,DA,9A,3A,57,01,CD,0933 [B8DC]
630 DATA 99E0,38,9A,10,FB,E1,3A,54,01,0845 [D572]
640 DATA 99E8,CD,9D,9A,3A,56,01,0D,20,0708 [85A0]
650 DATA 99F0,BD,01,06,00,F1,CA,27,9A,0832 [1E82]
660 DATA 99F8,01,05,00,C3,27,9A,E6,3F,0687 [E674]
670 DATA 9A00,CD,40,9A,E5,21,8D,9A,DD,1201 [3DC2]
680 DATA A376,42,F7,44,1F,02,47,F1,47,0797 [OC5A]
690 DATA A37E,47,F0,42,0F,02,40,F1,47,0770 [C150]
700 DATA A386,13,58,12,7C,90,28,24,12,0460 [C0EC]
710 DATA A38E,30,58,12,7C,90,28,24,12,0484 [C0C0]
720 DATA A396,DD,FF,7E,7F,FF,FE,FF,FB,1744 [71D4]
730 DATA A39E,FF,18,3C,7E,A5,7E,24,7E,0918 [34FE]
740 DATA A3A6,00,00,FF,00,00,00,00,00,0255 [8DC8]
750 DATA A3AE,00,18,34,6E,2C,18,00,00,0254 [7D20]
760 DATA A3B6,99,99,99,DB,FF,BD,99,FF,1530 [7248]
770 DATA A3BE,F7,99,99,99,DB,FF,BD,89,1530 [CD68]
780 DATA *ENDE* [BDD2]
790 adr=&A1C6:zeile=1040:MEMORY adr-1 [14C4]
800 READ d$:IF d$="*ENDE*"THEN 900 [C88C]
810 pr=0 [7810]
820 FOR i=1 TO 8 [286E]
830 READ a$:c=VAL("&*+a$) [F244]
840 POKE adr,a:adr=adr+1 [F320]
850 pr=pr+a [2894]
860 NEXT i [550C]
870 READ pr2 [D81E]
880 IF pr<>pr2 THEN PRINT"Pruefsummenfehler
in Zeile";zeile:STOP [1B20]
890 zeile=zeile+10:GOTO 800 [83CE]
900 SAVE"BDCK5.BIN",B,&A1C6,&200 [B01E]
910 PRINT d$:END [4D0E]
»BDCK6.BAS« (Schluß)
```