

# Große Buchstaben

**Wer hat in Titeln eigener Spiele nicht schon über die »normalen« Buchstaben geklagt? Mit diesem Programm wird das anders.**

Auf die Idee, dieses Programm zu verfassen, kam ich, als mir die immer gleich großen Buchstaben in den Titeln vieler Spiele so auf den Wecker gingen, daß ich selbst zum Programmierwerkzeug griff.

Das Programm stellt auf dem Bildschirm maximal 86, in der Höhe doppelt so große, Buchstaben dar. Diese können jeweils aus 1 bis 2 Farben bestehen.

Nun zum Programm selbst. In Zeile 5 bis 6 stehen die DATAs für das Maschinenprogramm, das den Bereich ab 12288 auf 0 setzt. Dadurch wird ein fließender Aufbau der Buchstaben garantiert. Die Zeilen 7 bis 10 lesen und starten das Maschinenprogramm. In Zeile 20 werden die Grundadresse des VIC sowie die neue Zeichensatz-Startadresse Variablen zugewiesen. Zeile 22 schaltet auf den neuen Zeichensatz um. Zeile 25 dient zum Setzen der Screen-Farben.

Die Variable »D« in Zeile 30 legt fest, ab welchem Zeichen undefiniert werden kann. Der Wert ist 82, da bis hier alle wichtigen und oft benötigten Zeichen, noch zum Auslesen nötig sind. Ab Zeile 60 beginnt dann das Setzen der umzudefinierenden Zeichen. Von Zeile 120 bis 180 werden dann die gesetzten Zeichen undefiniert und damit die neue Schrift auf dem Screen dargestellt. Das Ganze funktioniert wie folgt. Zuerst werden die alten Zeichen je viermal Byte für Byte ausgelesen. Jedes einzelne Byte wird dann zweimal dargestellt. Jetzt ist ein Zeichen umdefiniert. Nun wird der zweite Teil des Zeichens ausgelesen, verdoppelt und dem darunterliegenden Zeichen zugeordnet. In dem Unterprogramm ab Zeile 300 wird die Anfangsposition einer Druckzeile festgelegt.

Die DATAs für den Text können dann ab 500 stehen. Wenn dieses Programm aber später nur als Unterprogramm verwendet werden soll, empfiehlt es sich die DATA-Zeile ans Ende dieses Programms zu verlegen.

Eine DATA-Zeile ist folgendermaßen aufgebaut:

1. Spalte in der der Text später beginnen soll
2. Zeile
3. Anzahl der Zeichen die in eine Reihe sollen
4. Farbe obere Zeichenhälfte
5. Farbe untere Hälfte
6. -XX. Hier stehen die Bildschirmcodes der darzustellenden Zeichen.

Wichtig — als letztes Datum muß unbedingt fünfmal die -1 stehen. Da das Programm sehr kurz ist, ist es hervorragend zur Titelbildgestaltung von Spielen, Anwenderprogrammen oder ähnlichem geeignet.

(Matthias Baldauf/gr)

```

0 rem *****
1 rem *   big -- letters   *
2 rem *   1984 by mkb-soft *
3 rem * matthias baldauf 06361/7162 *
4 rem *   luitpoldstrasse 62 *
5 rem *   6760 rockenhausen *
6 rem *****
7 data 134,64,169,0,133,254,162,48,134,2
55,162,0,145,254,200,208,251
8 data 230,255,166,255,224,56,208,241,96
9 fort=52500to52500+25:reada:poket,a:nex
t
10 sys52500: rem * zeichen ram auf 0 dam
it fliesender aufbau
20 v=53248:x=12288:print"X":rem * x ist
startadresse des neuen zeichensatzes
22 pokev+24,29
25 pokev+32,0:pokev+33,0
30 d=82:rem * bildschirmcode ab dem zeic
hen undefiniert werden
40 y=x+82*8
55 gosub300
60 fori=1toza:rem * anzahl einzulesender
buchstaben
70 poke1,55:poke56334,1:q=1
80 forj=0to1:rem * 2 mal da zeichen jetz
t doppelt so hoch
85 ifd>255then400
90 pokec+i+j*40,d:rem * setzen des zeich
ens
100 pokee+i+j*40,fa(q):rem * setzen der
farben
110 q=q+1:d=d+1:next
120 reads
130 poke56334,0:poke1,51
140 forj=0to7:m=peek(53248+(s*8)+j):rem
* auslesen des alten zeichens (klein)
150 fork=0to1:rem * verdoppeln
160 pokey,m:y=y+1:nextk,j,i
170 poke1,55:poke56334,1
180 goto55
300 rem *** anfang setzen ***
301 :
305 readsp,ze,za,fa(1),fa(2):rem spalte,
zeile,anzahl der zeichen,farbe 1,farbe 2
306 ifsp=-1then400
310 c=1024+sp+ze*40:rem * wo text auf sc
reen steht * spalte und zeile*40
320 e=55296+sp+ze*40:rem * dasselbe im f
arbspeicher
330 return
400 rem *** einlesen beenden ***
401 :
402 end
403 rem ** hier geht spaeter programm we
iter
500 rem *** schrift datas ***
501 rem *** spalte * zeile * anzahl der
zeichen * farbe 1 * farbe 2 * schrift
502 :
503 data 10,4,19,7,10,42,32,20,8,5,32,2,
9,7,32,12,5,20,20,5,18,19,32,42
505 data 17,8,4,8,9,49,57,56,52
510 data 9,11,21,14,6,3,15,16,25,18,9,7,
8,20,32,2,25,32,13,11,2,45,19,15,6,20
520 data 8,19,23,4,4,22,9,5,12,32,19,16,
1,19,19,32,2,5,9,13,32,20,5,19,20
530 data 32,33,33
10000 data-1,-1,-1,-1,-1

```