Das Auge »ißt« mit



Bild 1. Ein interessantes Bild zieht das Auge an

Genauso wie beim Essen das Auge mitißt, steigert auch das optische Bild (Menü) eines Programmes das Interesse. Wie Sie diesen Appetitanreger in Ihr Programm einbinden können, erfahren Sie in diesem Kurs.

Viele Programme zeichnen sich dadurch aus, daß sie eine sehr anwenderfreundliche Form der Menüsteuerung (Bild 1) besitzen. Sieht man sich einige Menüs von Programmen kommerzieller Software-Hersteller an, wie zum Beispiel Textverarbeitungs- oder Dateiverwaltungsprogramme, so kann man hier mit den Cursortasten die einzelnen Programmfunktionen auswählen. Dabei werden mit den Cursortasten einzelne Menüpunkte hervorgehoben. Ist der gewünschte Menüpunkt erreicht, wird er mit einer Taste, zum Beispiel die F1-Taste, aktiviert.

Die einfachste Form der Menüsteuerung ist die, die einzelnen Menüpunkte durchzunumerieren. Mit der Basic-Anweisung »ON n GOTO« können die Unterprogramme sehr schnell aufgerufen werden. Zum Beispiel:

```
100 INPUT" MENUEPUNKT";A
110 ON A GOTO 1000,2000,3000,...,n
```

Eine solche Menüsteuerung wird in Listing 1 angewandt. Dieses Programm ermöglicht es, die Werte »SPIELSTUFE«, »GESCHWINDIGKEIT« und »ANZAHL DER SPIELER« zu verändern. Die Werte »SPIELSTUFE« und »GESCHWINDIGKEIT« können 1, 2 oder 3 annehmen, »ANZAHL DER SPIELER« 1 oder 2. Ein solches Menü könnte vor einem Spielprogramm stehen.

Über einen INPUT-Befehl (Zeile 320) wird die Nummer des gewünschten Menüpunktes eingegeben, auf Richtigkeit der Eingabe wird in Zeile 330 geprüft (es sind nur 1, 2, 3 oder 4 als Eingaben gestattet) und in der Zeile 360 in das Unterprogramm gesprungen. Der Befehl

```
360 ON P GOTO 380, 450, 520, 590
```

verzweigt, wenn die Variable P, die in Zeile 320 die Nummer des gewünschten Menüpunktes vom Benutzer über die INPUT-Anweisung (320 INPUT ".. ";P) erhält, den Wert 1 hat, nach 380, bei P=2 nach 450 und so weiter.

Die Adressen, die hinter dem ON n GOTO-Befehl stehen, zeigen jeweils auf die erste Zeilennummer einer Unterroutine, die vorher per Menü ausgewählt wurde. Zu den Zeilen 370 bis 620 ist nichts weiter zu sagen, als daß dort die Unterroutinen stehen, die die Eingabe von Werten beziehungsweise das Verlassen des Menüs bewirken.

Die Variablen L, G und A enthalten die Werte für »Spielstufe« (<L> für »Level«), »Geschwindigkeit« <G> und »Anzahl der Spieler« <A>. Die Variable P enthält die Nummer des Menüpunktes, der ausgeführt werden soll.

Diese Form des Menüs, die in Listing 1 verwendet wird, ist allerdings noch recht unkomfortabel. Störend ist, daß jede Eingabe der Menünummer immer mit der RETURN-Taste bestätigt werden muß. Im folgenden soll dies geändert werden. Dies erreicht man durch Verwendung einer GET- beziehungsweise GETKEY-Abfrage anstelle des INPUT-Befehls, wie in Listing 2 in Zeile 640 zu sehen ist. Listing 2 hat jedoch noch eine weitere kleine Ergänzung, die ich Ihnen hier nebenbei vorstellen möchte, weil sie simpel, somit leicht verständlich und ohne nennenswerte Probleme programmierbar und wirkungsvoll zugleich ist: die Window-Technik.

Damit Sie sofort erkennen können, worum es sich handelt, sollten Sie Listing 2 abtippen, starten und alle Funktionen ausprobieren. Jetzt drücken Sie nur noch 1, 2, 3 oder 4 (je nach Menüpunkt), dann können Sie gegebenenfalls einen

der Werte »Spielstufe«, »Tempo« (=»Geschwindigkeit« aus Listing 1) und »Anzahl der Spieler« ändern. Diese verbesserte Form der Parametereingabe hat allerdings nichts mit der Window-Technik zu tun, mit der wir uns nun beschäftigen wollen.

Ein Window (»window« = englisches Wort für »Fenster«) ist ein Bildschirmausschnitt, der als Teilbildschirm behandelt wird und für einen bestimmten Zweck (zum Beispiel die Anzeige eines Copyrights) reserviert ist. Mit Hilfe der Sequenzen ESC-B beziehungsweise ESC-T können Sie beim C 16/116 das Window bestimmen, es handelt sich jedoch nur um eine Verkleinerung des Bildschirmbereiches: dieser Teilausschnitt wird wie der gesamte Bildschirm gehandhabt. Dies ist nützlich, um Teile des Bildschirms vor dem Überschreiben zu schützen (für eigene Programme äußerst unbrauchbar). Im Beispielprogramm (Listing 2) sind einzelne Teile des Bildschirms für bestimmte Aufgaben freigehalten; so gibt es:

- ein Copyright-Window, in dem der Copyright-Vermerk steht,
- ein Level-Window, in dem die Spielstufe steht,
- ein Tempo-Window, in dem die Geschwindigkeit steht,
- und ein Anzahl-Window, in dem die Anzahl der Spieler steht.

Die Windows werden in den Zeilen 290 bis 590 ausgegeben, und zwar dann, wenn auf eine neue Eingabe gewartet wird. Da die Windows immer an derselben Bildschirmposition stehen und nicht verschoben werden, muß vor der Neuausgabe eines Windows nicht erst der Bildschirm gelöscht werden. Dieser Trick, immer nur deckungsgleiche Bereiche auszugeben (wodurch dem Benutzer der Neuaufbau eines Bildschirmbereichs nicht auffällt), wird an späterer Stelle in diesem Kurs – bei einer anderen Form der Menüsteuerung – erklärt. Dort können Sie dann die Funktionsweise erfahren, die uns hier im Moment nicht weiter aufhalten soll.

Bei der Verwendung von Windows sind folgende Punkte zu beachten:

1. An der Stelle, an der ein Window stehen soll, darf vorher kein anderer Text stehen, da dieser sonst durch das Window überschrieben wird. Wenn Sie einmal sehen wollen, wie ein Window einen Teil des Bildschirms überschreibt, ändern Sie jeweils die drei Parameter nach CHAR in den Zeilen 340 bis 390 von 2 bis 7 auf 7 bis 12. Starten Sie dann das Programm, und das Copyright-Window überschreibt die Menüpunkte teilweise.

2. Windows muß man optisch abgrenzen, damit sie vom »normalen« Bildschirm unterschieden werden können. In der Regel nimmt man Grafikzeichen, die das Window einrahmen. In Listing 2 sehen Sie einige verschiedene Möglichkeiten der Eingrenzung eines Windows (durch Reversdruck, Grafikzeichen, Sterne), und es gibt noch mehr Variationen.

3. Die Position des Windows muß festgelegt sein. Vor dem Ausdrucken eines Windows muß diese Window-Position als Cursorposition eingestellt werden. Hierbei ist der CHAR-Befehl eine große Hilfe. Dieser Befehl kann nämlich nicht nur Texte in hochauflösende Grafik schreiben, sondern auch wie ein PRINT-Befehl bestimmte Stellen im normalen Text-Modus beschreiben. Die Syntax für CHAR zum PRINTen an eine beliebige Cursorposition ist:

```
CHAR 1, Spalte,Zeile,"TEXT ALS STRING"
```

Die »1« hinter CHAR gibt an, daß in die normale Textseite geschrieben werden soll. Spalte und Zeile entspricht der Cursorposition, bei der die Textausgabe beginnen soll. »TEXT ALS STRING« ist der auszugebende Text.

Ein Tip: In der Regel wird man den Wert für »Spalte« beim Drucken eines Windows nicht ändern, da der linke Rand nicht verändert wird; der Wert für »Zeile« ist jedoch dauernd um 1 hochzuzählen, wie man unter anderem an den Zeilen 340 bis 390 verfolgen kann (3. Parameter nach CHAR!).

4. Der Einfachheit halber sollte man nur Windows in Rechteckform programmieren, weil man in diesem Fall nur zwei Begrenzungen des Windows kennen muß: Länge und Breite. Da das C 16/116-Betriebssystem das Arbeiten mit Windows, wie wir es benötigen würden, nicht unterstützt, muß man auf die Einhaltung dieser Grenzen selbst achten. Bei Rechteck-Windows der gebräuchlichen Form muß man folgende Werte festlegen:

- linke Grenze
- rechte Grenze
- obere Grenze
- untere Grenze

Man darf also, weil ein Window kleiner als der gesamte Bildschirm ist, nur eine begrenzte Textbreite verwenden; außerdem stehen nicht beliebig viele Zeilen zur Verfügung. Dies ist ganz einfach eine Frage der Planung beim Schreiben der PRINT- beziehungsweise CHAR-Zeilen. Betrachten wir dazu einmal die Zeilen 340 bis 390. Jeder dieser sechs CHAR-Befehle hat als 2. Parameter eine 5, das Drucken wird also immer in der linken Spalte begonnen. Damit ist die Einhaltung der linken Grenze garantiert. Die obere Grenze wird dadurch eingehalten, daß die Ausgabe in der 2. Zeile beginnt (Zeile 340: CHAR .,.,2,...), die untere Grenze dadurch, daß nicht weiter als bis zur 7. Zeile gedruckt wird (Zeile 390: CHAR .,.,7,...).

Die rechte Grenze einzuhalten ist am schwierigsten, denn hier kommt es nicht nur auf die Parameter des CHAR-Befehls an. Wenn man aber eine einheitliche Länge des Ausgabe-Strings einhält – in den Zeilen 340 bis 390 hat jeder Ausgabe-String exakt 11 Zeichen Länge, das Window ist also 11 Zeichen breit – dürfte es keine Probleme geben. Wir können nur abschließend wiederholen: Wenn man beim Programmieren ausreichend genau vorgeht, sind Windows ein Kinderspiel.

Wir hoffen, daß Ihnen der Ausflug ins Thema »Windows« Spaß gemacht hat; nun wollen wir uns bis zum Ende dieses Kurses mit einer ähnlich ästhetischen Form der Menüsteuerung befassen. Dazu ist zunächst etwas vorauszuschicken.

Der C 16/116 bietet im Gegensatz zum VC 20/C 64 eine Funktionstastenbelegung, die Sie als C 16/116er sicher zu schätzen wissen. Der einzige Haken dieser Funktionstastenbelegung ist jedoch, daß die Abfrage der Funktionstasten über GET/GETKEY erschwert wird. Um dies zu umgehen, ist es erforderlich, den C 64/VC 20-Zustand wiederherzustellen, indem die Funktionstasten wie Steuertasten (CLEAR, HOME, Cursortasten, Farbtasten,...) gehandhabt werden. Zwar kann man mit den auf die Funktionstasten gelegten Steuertasten keine Steuerfunktion hervorrufen (die verwendeten Codes 133 bis 140 haben eben keine Funktion), aber man kann leicht auf eine Funktionstaste warten:

```
10 KEY 1,CHR$(133):REM F1-Taste mit Code belegen
20 GETKEY A$:IF A$<>CHR$(133) THEN 20:REM auf F1-
Code warten
```

Von den Steuertasten wie CLEAR, HOME, Cursortasten und so weiter kennen Sie die Möglichkeit, Steuerzeichen in Anführungszeichen als reverse Symbole zu schreiben (siehe Eingabehinweise für Basic-Programme in diesem Heft). Nach der Belegung der Funktionstasten mit Steuercodes geht dies auch mit F-Tasten. Nach Eingabe von KEY 1,CHR$(133) ist die F1-Taste mit dem Code 133 belegt; durch einmaliges Drücken der Anführungszeichen-Taste wird die F1-Taste als reverses Grafikzeichen abgebildet, wie es innerhalb von Anführungszeichen auch mit der HOME-Taste geschieht.

Alle nun folgenden Listings machen von der Abfrage der F1- und der F7-Taste (die anderen Funktionstasten bleiben unberührt) Gebrauch. Dadurch steht zwar auf F1 nicht mehr »GRAPHIC« und auf F7 nicht mehr »LIST«, aber wir können die Tasten abfragen. Die erforderlichen KEY-Befehle stehen in jedem der nun folgenden Listings, müssen jedoch schon



Bild 2. Das Menü aus Listing 3

vor der Eingabe der Programme eingegeben werden, um in Anführungszeichen die F1/F7-Taste mit einem Steuerzeichen auszudrücken. Geben Sie also vor der Eingabe aller folgenden Listings dieses Kurses die Anweisung

```
KEY 1,CHR$(133):KEY 7,CHR$(136)
```

im Direktmodus ein. Wenn Sie ein solches gespeichertes Programm laden wollen, sind diese Befehle nicht erforderlich, da sie bereits im Programm stehen.

Damit Sie gleich in den Genuß eines guten Menüs kommen, tippen Sie einfach Listing 3 ab und starten es. Der Bildschirm wird dunkel und ein Menü mit den Punkten A bis E erscheint. Hinter jedem Menüpunkt steht eine Zahl in Klammern, die die Nummer des Unterprogramms angibt, das Sie anwählen können. Die Funktionen des kleinen Programms im einzelnen:

Mit <CURSOR DOWN> kommen Sie zum nächsten Punkt.

Mit <CURSOR UP> erreichen Sie den vorhergehenden Punkt.

Vom untersten Menüpunkt aus können Sie mit CURSOR DOWN direkt in den obersten und mit CURSOR UP vom obersten Punkt zum untersten springen.

Mit HOME können Sie von jedem Punkt immer in den obersten springen.

<F1> aktiviert das angewählte Unterprogramm.

<F7> bricht das Programm mit END ab.

Probieren Sie ohne weiteres alle Funktionen von Listing 3 aus (Bild 2). Sollte Ihnen diese Art der Menüsteuerung gefallen haben, dann erfahren Sie im folgenden mehr über den Aufbau.

Im Bild 3 finden Sie die schematische Darstellung der Menüpunkte von Listing 3. Die Randpunkte A (0) und E (4) sind besonders hervorgehoben, weil an diesen die Rotation stattfindet.

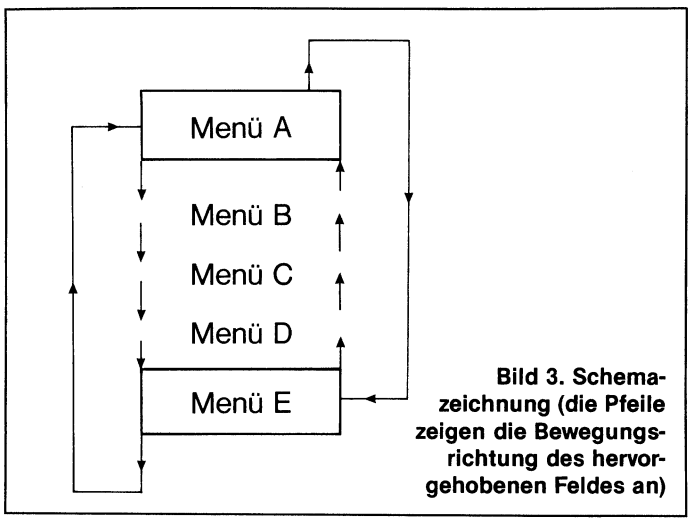

Bild 3. Schemazeichnung (die Pfeile zeigen die Bewegungsrichtung des hervorgehobenen Feldes an)

Die Variablen von Listing 3:

A (Menüanzahl) enthält die Anzahl der Menüpunkte (5). A ändert sich nie und ist folglich eine Konstante.

R$(n) ist immer ein 1-Byte-String, der den Wert CHR$(0) enthält. Der Code CHR$(0) steht für Null und hat keine Funktion. Wenn ein Menüpunkt hervorgehoben is, so steht in R$(n) der Wert CHR$(18) für <REVERS ON>.

P (Programmpunkt) enthält die Nummer des angewählten Unterprogramms. P wird durch die Cursortasten geändert und gibt den Menüpunkt an, der hervorgehoben werden soll. Des weiteren wird P zur Anwahl der Unterprogramme benötigt. (490 ON P+1 GOTO 610, 680, ...)

T$ (Taste) In der Zeile 480 wird die Tastatur abgefragt. Der Tastencode wird in der Variablen T$ gespeichert. In den Zeilen 490 bis 500 und 520 bis 540 wird der Tastencode mit Bedingungen verglichen. Ist die Bedingung wahr, so wird die nachfolgende Anweisung ausgeführt.

Nachdem die Variablen beschrieben wurden, soll nun der Programmaufbau betrachtet werden.

Zeile 200 bis 300

In der Zeile 250 werden die Variablen definiert. Nach der DIM-Anweisung in der Zeile 250 ist übrigens das gesamte Feld R$(n) mit Leerstrings belegt. (Sollten Ihnen einige Begriffe, zum Beispiel »Feld« nicht bekannt sein, so schauen Sie sich doch bitte den Basic-Kurs oder das Lexikon an.)

In der Zeile 260 wird die Variable P (Programmpunkt) auf Null gesetzt (Menüpunkt A).

Die Anweisung R$(P) = CHR$(18) in Zeile 270 hebt den Menüpunkt hervor, indem der Zeichenhintergrund die Farbe des Zeichens erhält und das Zeichen in Schwarz dargestellt wird. Im folgenden werde ich diese Darstellung als invers bezeichnen.

Schließlich werden in Zeile 280 die Bildschirm- und Rahmenfarbe auf Schwarz gesetzt, der Groß-/Grafikmodus aktiviert und der Bildschirm mit PRINT CHR$(147) gelöscht.

Hauptschleife: Zeile 310 bis 560

Bei der Zeile 310 beginnt die Hauptschleife. Das Menü springt immer von Zeile 560 aus auf 310 zurück, solange kein Unterprogramm aktiviert wurde.

Textausdruck: Zeile 340 bis 420

In den Zeilen 340 bis 420 findet man die Textausgabe der fünf Menüpunkte. (Die Zeile 340 wird anschließend noch gesondert besprochen). Vor den einzelnen Menüpunkten steht die Stringvariable R$(n):

```
380 PRINT TAB(9);R$(0);"{Farbe}Menuepunkt A {Revers
off}(0){CURSOR DOWN}"
```

R$(n) wirkt hier als Schalter. Ist R$(n)=CHR$(0), so schaltet er den Reversmodus nicht ein. Ist dagegen R$(n)=CHR$(18), so schaltet er den Modus ein. Die Zeile 430 gibt das gerade angewählte Programm (Nummer) an.

Warten auf Tastendruck: Zeile 450 bis 480

Diese Programmzeile holt ein Zeichen von der Tastatur und legt sie in der String-Variablen T$ ab. Trifft die Bedingung T$ = Leerstring zu, so verzweigt das Programm zurück zum Anfang dieser Zeile. Damit wartet das Programm in dieser Schleife, bis eine Taste gedrückt wurde.

```
480 GET T$ : IF T$ = "" THEN 480
```

Programmaufruf: Zeile 490

In der Zeile 490 wird geprüft, ob die zuletzt gedrückte Taste die Funktionstaste F1 war. Ist diese Verknüpfung wahr, so wird die Anweisung, die hinter THEN steht, ausgeführt.

```
490 IF T$="{F1}" THEN ON P + 1 GOTO 610, 680, 730
```

Dabei bedeutet die Anweisung ON P GOTO (Zeile): springe auf die Zeilennummer, die durch den Zeiger P angewählt ist. Ein Beispiel: P enthält den Wert 2. Zu diesem wird laut Anwei-

sung eine 1 addiert (P + 1) und ist damit 3. Dieser Wert gibt die Position (3) in der Tabelle an, an der die Zeilennummer steht, zu der gesprungen werden soll. In unserem Beispiel wäre es die Zeile 730.

Programmabbruch: Zeile 500

Ergibt die Prüfung `T$ = "{F7}"`, daß diese Bedingung zutrifft, so wird das Programm mit einer Bemerkung beendet.

```
500 IF T$ = "{F7}" THEN PRINT "{CURSOR 2 mal DOWN}
Programmabbruch." : END
```

Rücksetzen: Zeile 510

Es wird die Stringvariable R$(P), die an dieser Stelle noch den Wert CHR$(18) enthält, auf CHR$(0) geändert.

```
510 R$(P) = CHR$(0)
```

Programmanwahl: Zeile 520 bis 540

In diesen drei Zeilen wird geprüft, ob die Taste CURSOR UP, CURSOR DOWN oder HOME gedrückt wurde. (Auf diese Anweisungen wird noch gesondert eingegangen.)

Menüpunkt invertieren: Zeile 550

In der Zeile 550 wird der Variablen R$(P), wobei P jetzt den neuen Menüpunkt beinhaltet, der Wert CHR$(18) zugewiesen.

```
550 R$(P) = CHR$(18)
```

Rücksprung: Zeile 560

Hier ist die Hauptschleife beendet und es wird auf die Zeile 310 zurückgesprungen.

```
560 GOTO 310
```

Unterprogramme: ab 570

Es folgen die fünf Unterprogramme, die Sie vom Menü aus aufrufen können.

Die Struktur des ersten Programms wäre damit, bis auf einige Besonderheiten, beschrieben. Um die Bewegung des Reversfeldes zu simulieren, muß das alte Bild bei jedem Tastendruck mit dem neuen Bild überschrieben werden. Die Frage ist: wie erreicht man es, daß sich die Bilder absolut genau decken? Man legt dazu einen Bezugspunkt, die linke obere Ecke, auf dem Bildschirm fest. Der Basic-Befehl dazu ist HOME oder PRINT CHR$(19). Fügt man den Befehl als PRINT CHR$(19)-Anweisung in ein Programm ein, so springt der Cursor in die linke obere Ecke des Bildschirms. Läßt man nun den Bildschirminhalt erneut ausdrucken, so liegt er deckungsgleich zum alten Bild. Die Anweisung PRINT CHR$(19) kann auch durch die Sequenz, PRINT"{HOME}", ausgedrückt werden.

Ersetzen Sie in der Zeile 340 die Anweisung CHR$(19) durch CHR$(147) (Bildschirm löschen), so wird der Neuaufbau des Bildes deutlicher, weil der Bildschirm jedesmal komplett gelöscht wird. Dies macht sich dann durch ein Flackern des Bildes bemerkbar.

Die eigentliche Besonderheit des kleinen Programms steht in den Zeilen 520 bis 540. Mit der Zeile 540, der einfachsten, soll begonnen werden.

```
540 IF T$ = "{HOME}" THEN P = 0
```

Ist die Bedingung `T$ = "{HOME}"` wahr, so wird die Variable auf 0 gesetzt. Damit enthält P die Nummer des obersten Menüpunktes und druckt diesen in inverser Schrift aus.

Betrachtet man noch einmal Bild 3, so erkennt man, daß das Inversfeld jeweils um 1 nach oben wandert, wenn die CURSOR UP-Taste gedrückt wird. Zum Beispiel wandert das Feld von C nach B und der Wert für P sinkt um 1, von 3 auf 2. Daher heißt es auch in der Zeile:

```
530 IF T$ = "{CURSOR DOWN}" THEN P = P - 1 ...
```

Hatte die Variable vorher den Wert 0, so vermindert sie sich jetzt auf den Wert (-1). Ein Menüpunkt mit der Nummer (-1) existiert aber nicht, statt dessen soll vielmehr von dem Menüpunkt A aus direkt nach Menü E gesprungen werden. Es folgen nun mehrere Möglichkeiten, mit denen diese Bedingung erfüllt werden kann. Ein entsprechendes Flußdiagramm finden Sie im Bild 4.

Der erste Weg:

```
P = P - 1 : IF P = (-1) THEN P =
```

oder

```
P = P - 1 : IF P = (-1) THEN P = P + 5
```

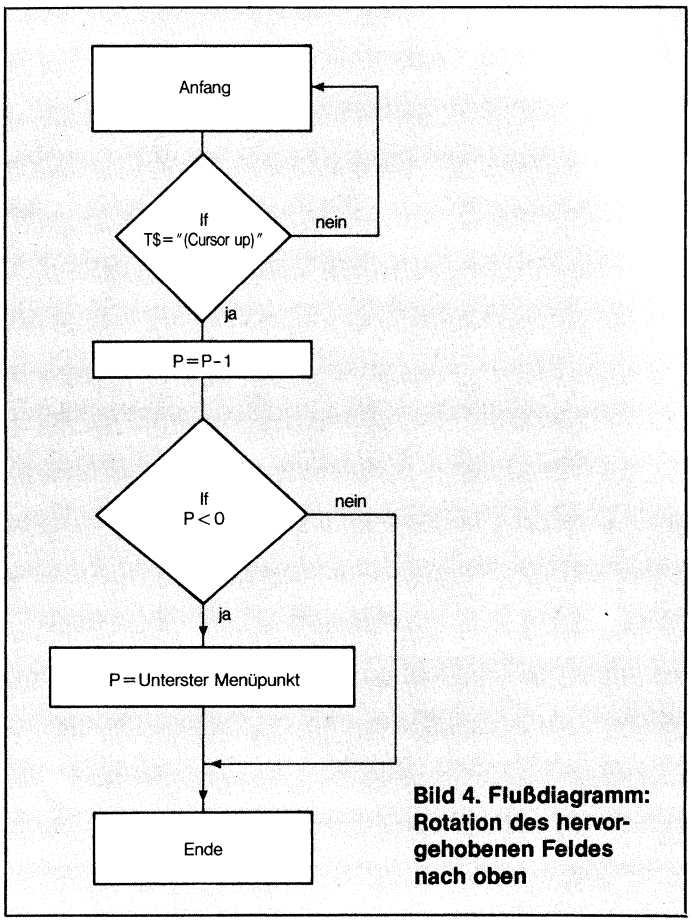

Bild 4. Flußdiagramm: Rotation des hervorgehobenen Feldes nach oben

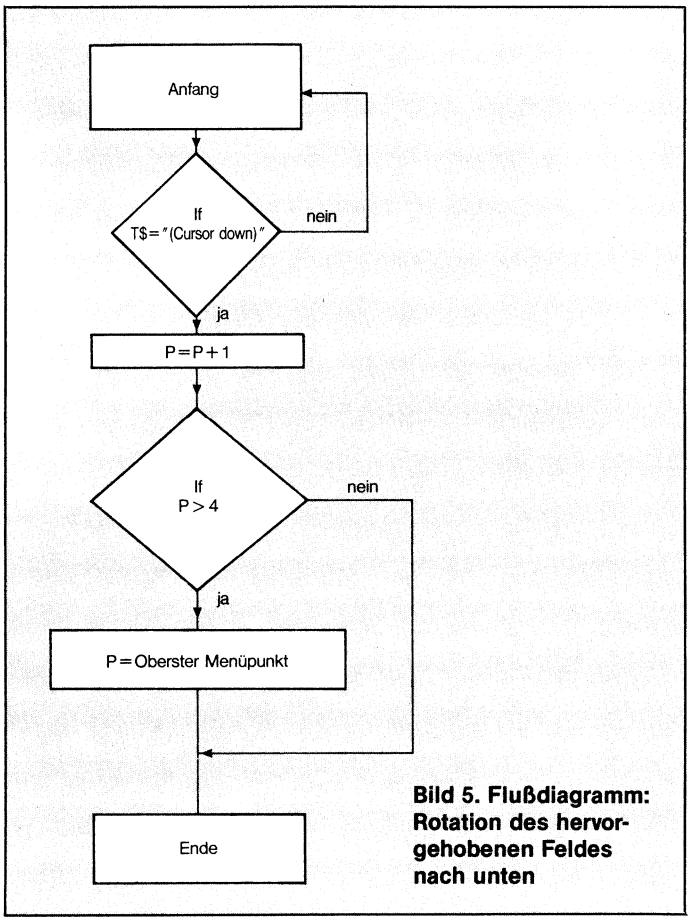

Bild 5. Flußdiagramm: Rotation des hervorgehobenen Feldes nach unten

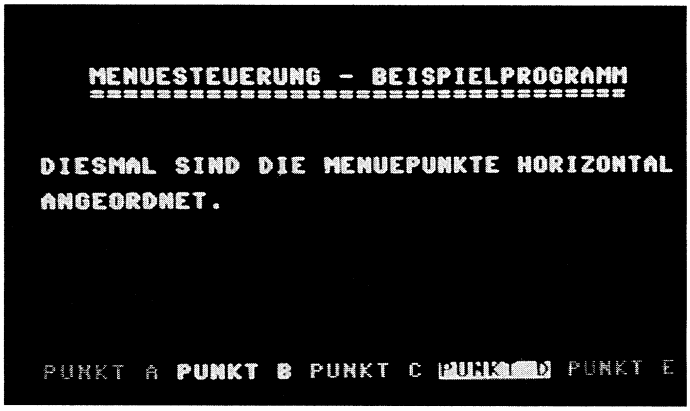

Bild 6. Das horizontale Menü aus Listing 5

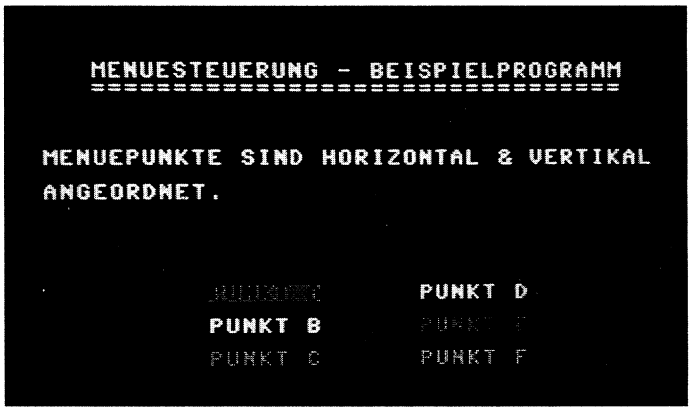

Bild 7. Das zweidimensionale Menü aus Listing 6

Der Weg, der beschritten wurde, ist eigentlich deutlich. Es wird das Ergebnis aus der Subtraktion P - 1 mit dem Wert (-1) verglichen. Ergibt der Vergleich, daß P = (-1) ist, dann wird der Wert der Variablen P durch 4 ersetzt oder um 5 vermehrt.

In der Zeile 530 steht dagegen:

P = P - 1 - A * (P = 0)

Hier wird zunächst auch der Wert der Variablen P um 1 vermindert. Ist die in Klammern stehende Bedingung (P = 0) wahr, das heißt gleich, so wird der Ausdruck durch (-1) ersetzt. Dieser Wert kann nun in die Rechnung einbezogen werden. Die Gleichung lautet dann:

```
   ...THEN P = P - 1 - A * (P = 0)
=→...THEN P = P - 1 - A * (-1)
=→...THEN P = P - 1 - (-A)
=→...THEN P = P - 1 + A
```

Ersetzen Sie die Konstante A durch 5 und die Variable P durch 0, so erhalten Sie folgende Gleichung:

```
   ...THEN P = 0 - 1 - 5 * (0 = 0)
=→...THEN P = 0 - 1 - 5 * (-1)
=→...THEN P = 0 - 1 - (-5)
=→...THEN P = 0 - 1 + 5
=→...THEN P = 4
```

Ist die Bedingung P = 0 nicht wahr, so wird vereinbarungsgemäß eine 0 für die Antwort »das ist falsch« geschrieben. Ersetzt man in der Rechnung den Ausdruck (P = 0) durch unwahr (0), so ändert sich die Gleichung in

```
   ...THEN P = P - 1 - A * (P = 0)
=→...THEN P = P - 1 - A * (0)
=→...THEN P = P - 1 - 0
=→...THEN P = P - 1
```

Kontrollieren Sie diese Gleichung wieder mit folgenden Werten: A = 5 und P = 2.

Betrachtet man nun die Zeile 520, so ähnelt sie der Zeile 530. Deshalb kann die Abhandlung auch etwas kürzer ausfallen (Bild 5).

In der Zeile 520 steht:

...P = P + 1 + A * (P = (A - 1))

Auch hier ist wieder die Verknüpfung (P = (A-1)) der Hauptbestandteil. Geht man davon aus, daß P = 4 ist, so muß die Rechnung wie folgt aussehen.

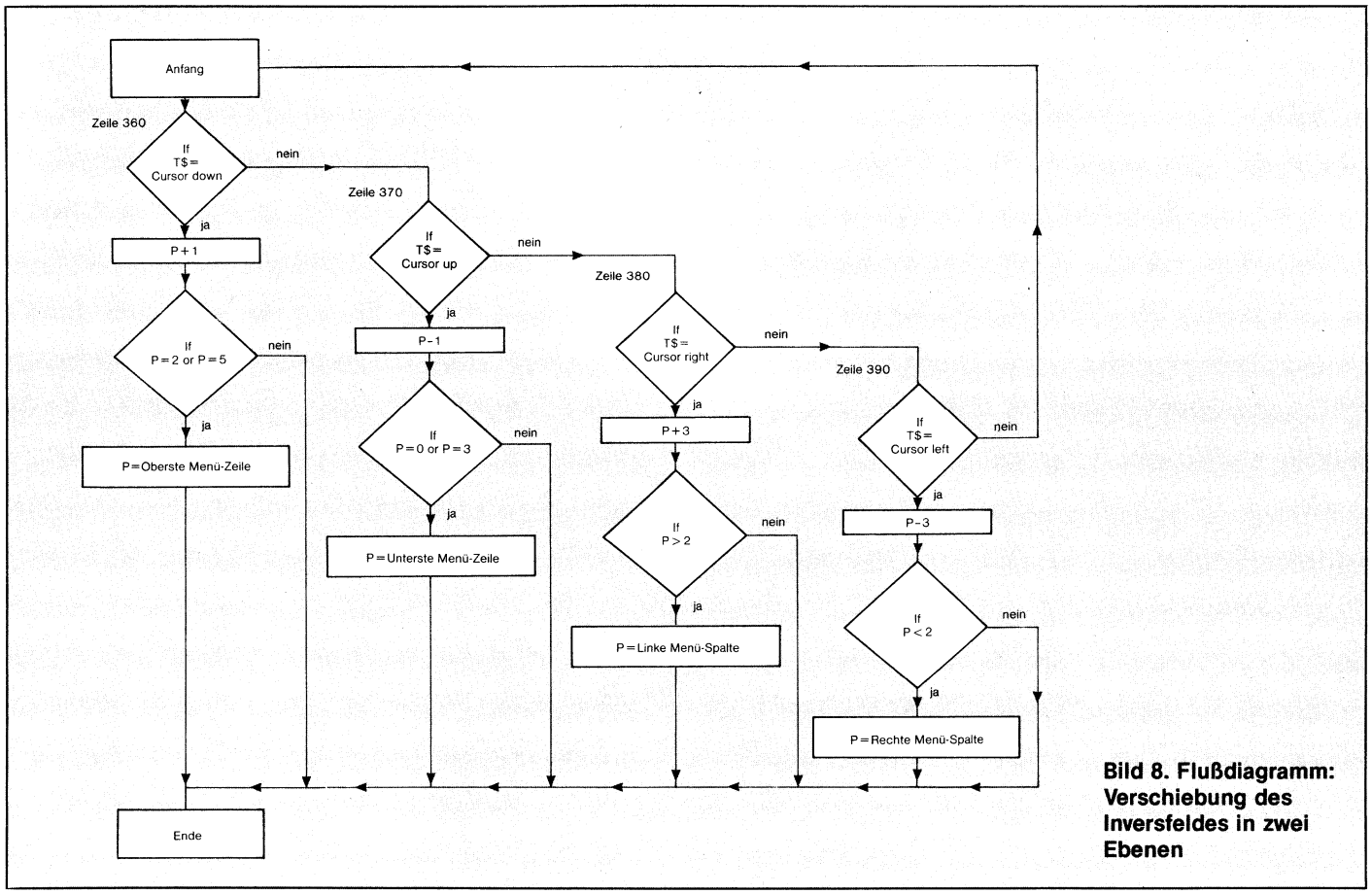

Bild 8. Flußdiagramm: Verschiebung des Inversfeldes in zwei Ebenen

```
   ...THEN P = P + 1 + A * (P = (A - 1))
=→...THEN P = P + 1 + A * (-1)
=→...THEN P = P + 1 + (-A)
```

Da P = 5-1 wahr ist, sieht die mit Zahlen ersetzte Gleichung so aus:

```
   ...THEN P = 4 + 1 + 5 * (4 = (5 - 1))
=→...THEN P = 4 + 1 + 5 * (-1)
=→...THEN P = 4 + 1 + (-5)
=→...THEN P = 0
```

Ich nehme an, daß Ihnen der Algorithmus klar geworden ist. Im Anhang finden Sie noch weitere Erklärungen zur Computerlogik.

Warum nun diese umständliche Schreibweise? Alle IF-Anweisungen benötigen viel Rechenzeit, mathematische Operationen dagegen sind verhältnismäßig schnell. Deshalb lohnt sich die Verwendung dieses Algorithmus in allen Vergleichsverfahren, die durch eine Rechnung ersetzt werden können.

In der untersten Zeile unseres Menüs steht die Anmerkung »Derzeit angewählter Menüpunkt«. Dahinter wurde bisher die Nummer des Menüpunktes angezeigt. Dagegen sind alle Menüpunkte im Text durch einen Buchstaben gekennzeichnet (Bild 2). Deshalb wird die Zeile 430 geändert in:

```
430 PRINT "DERZEIT ANGEWAEHLTER MENUEPUNKT:" CHR$
(65 + P)
```

Der Code für den Buchstaben »A« ist die 65 (siehe auch im Commodore-Handbuch ASCII-Code). Alle im Alphabet folgenden Buchstaben haben jeweils eine um 1 größere Codezahl. Auf Grund dieser Tatsache läßt sich der Buchstabe des Menüpunktes mit der kleinen Rechnung

Code = 65 + P

ermitteln.

Listing 2 ist eine Variante von Listing 3, wie sie in der Praxis angewendet wird. Die Informationen bezüglich der internen Numerierung sowie die REM-Zeilen wurden weggelassen. Außerdem wurden die Zeilen teilweise komprimiert. So steht für PRINT CHR$(0) jetzt PRINT" " und für PRINT CHR$(18) jetzt PRINT"{Steuertaste}". Die wichtigste Änderung dagegen ist, daß die Konstante A entfällt und dafür alle Ausdrücke, in denen A vorkommt, ausgerechnet werden. (Vergleichen Sie die Zeilen 520 und 530 von Listing 3 mit 360 und 370 von Listing 4.)

Warum nicht horizontal?

Bislang wurden die Menüs untereinander angeordnet, man könnte auch von einem vertikalen Menü sprechen. Es ist aber kein Problem, die Menüs auch horizontal (waagerecht) anzulegen. In Bild 6 sehen Sie, wie dies gemeint ist. An den Punkten A und E erfolgt wieder die Rotation. Bei einem horizontalen Menü entspricht die Bewegung des Inversfeldes nach rechts die der Bewegung nach unten im vertikalen Menü. Die Bewegung nach links entspricht der nach oben. In Listing 5 finden Sie ein horizontales Menü, wie Sie es in Bild 6 sehen. Die mit der Taste F1 aufgerufenen Unterprogramme geben den Inhalt der Variablen P an. Wie Sie in Zeile 360 und 370 sehen, wird die Funktion der CURSOR DOWN-Taste von der CURSOR RIGHT-Taste übernommen. Ebenso ersetzt die CURSOR LEFT-Taste jetzt die CURSOR UP-Taste. Der größte Teil von Listing 5 wurde aus dem Listing 4 übernommen. Außer der Tastaturabfrage haben sich nur die Printanweisungen geändert. Zusätzlich sind die Zeilen 410 bis 580 (Unterprogramme) durch die Anweisung

```
330 IF T$ = "{F1}" THEN PRINT "ANGEWAEHLTER PUNKT:"; P
: END
```

ersetzt worden.

Zum Abschluß des Kurses soll noch das vertikale und horizontale Menü miteinander verbunden werden. Es entsteht ein zweidimensionales Menü (Listing 6). In den mehrdimensionalen Menüs werden alle vier Cursortasten mit einbezogen. Die CURSOR UP- und -DOWN-Tasten werden im Prinzip wie im vertikalen Menü (siehe Listing 3) zur Ansteuerung der vertikalen Ebenen (A-D, B-E, C-F siehe Bild 7) verwendet. Die CURSOR LEFT- und -RIGHT-Tasten werden zur Anwahl der beiden Spalten verwendet. Die Flußdiagramme finden Sie in Bild 8. Die entsprechenden Berechnungen stehen in den Zeilen 360 bis 390.

```
360 IF T$ = "{CURSOR DOWN}" THEN P = P + 1 + 3*
(P = 2 OR P = 5)
370 IF T$ = "{CURSOR UP}" THEN P = P - 1 - 3 *
(P = 0 OR P = 3)
380 IF T$ = "{CURSOR RIGHT}" THEN P = P + 3 + 6
(P >2)
```

Logik-Befehle

In Basic gibt es verschiedene Möglichkeiten Vergleiche durchzuführen. Jeden Vergleich können wir auch als Frage ansehen. So zum Beispiel könnte man für den Vergleich »A = B« fragen:

Ist der Wert A ebensogroß wie der Wert B? Als Antwort stände: »das ist wahr« oder »das ist falsch«. Nach diesem Prinzip behandelt auch der Computer die Vergleiche. Wenn man die Antwort jedesmal in einem Satz ausgeben würde, bräuchte man viel Platz. Deshalb hat man sich geeinigt, für den Ausdruck »das ist wahr« (-1) anzugeben und für »das ist falsch« (0). Diese Werte ((-1) und 0) können wir auch zum Rechnen verwenden. In der folgenden Tabelle finden Sie alle Vergleichsbefehle mit ihrer Beschreibung.

=	sind die Werte gleich groß?
<	ist der links stehende Wert kleiner als der rechts stehende?
>	ist der links stehende Wert größer als der rechts stehende?
<>	sind die Werte verschieden groß?
<=	ist der links stehende Wert kleiner oder gleichgroß mit dem rechts stehenden?
>=	ist der links stehende Wert größer oder gleichgroß mit dem rechts stehenden?

Alle in dieser Tabelle stehenden Bedingungen (Vergleiche) können nur mit wahr oder falsch beantwortet werden. Es gibt aber noch eine Reihe anderer Logik-Befehle, wie OR oder AND. Diese unterscheiden sich von den ersten Logik-Befehlen dadurch, daß hier mehrere Antworten möglich sind. Ein Beispiel:

A = (27 AND 8) => A = 8

Warum ist das Ergebnis acht? In unserem Beispiel wird der erste Wert binär mit dem zweiten Wert verglichen.

```
   27 = 0001 1011
AND 8 = 0000 1000
-----------------
    8 = 0000 1000
```

Verknüpfen wir die Bits jeder Spalte, von rechts nach links, nach folgendem Algorithmus: 0 AND 1 oder 1 AND 0 = 0 und 1 AND 1 = 1 . Das entspricht also einer einfachen Multiplikation. Das heißt in unserem Beispiel, daß die ersten vier Bit zu Null werden. In der 5. Spalte steht jeweils in der ersten und zweiten Zeile eine 1 und somit ist die Verknüpfung auch eine 1 (1 mal 1 = 1).

Mit dieser Verküpfung können wir bestimmte Werte (binär) aus einem anderen Wert herausfiltern.

Beispiel:

A = (46 AND 15). Hier werden die Wertanteile von 0 bis 15 herausgefiltert.

```
   46 = 0010 1110
AND 15 = 0000 1111
-----------------
   14 = 0000 1110
```

Die andere Verknüpfung ist die OR-(Oder-)Verknüpfung. Hier lautet der Algorithmus: ist entweder der eine Wert oder der andere oder beide Werte 1, so ist das Ergebnis 1, sind beide Werte 0, so ist auch das Ergebnis 0.

Ein Beispiel:

```
   13 = 0000 1101
OR 46 = 0010 1110
-----------------
   47 = 0010 1111
```

Mehr über IF-Abfragen und logische Vergleiche finden Sie im Basic-Kurs.

```
390 IF T$ = "{CURSOR LEFT}" THEN P = P - 3 - 6
(P <3)
```

Da nur zwei Spalten vorhanden sind, müßte nicht zwischen CURSOR RIGHT und CURSOR LEFT unterschieden werden. Beide Tasten veranlassen nur den Wechsel in die andere Spalte. Eigentlich ließe sich dieser Sonderfall durch eine OR-Verknüpfung einfacher realisieren. Sie lautet:

```
380 IF T$ = "{CURSOR RIGHT}" OR T$ =
"{CURSOR LEFT}" THEN ...
```

Auf die THEN-Anweisung könnte jede der Formeln aus Zeile 380 oder 390 passen.

Als kleine Beigabe habe ich für Sie noch eine JA/NEIN-Abfrage (Listing 5). Im Prinzip haben Sie hier wieder ein horizontales Menü. Mit den Cursortasten können Sie eine der beiden Antworten anwählen. Durch Drücken der F1- oder RETURN-Taste veranlassen Sie die Ausführung des Unterprogramms für Ja oder Nein. (Florian Müller/do)

```
100 REM ********************************
110 REM *                              *
120 REM * B E I S P I E L - M E N U E *
130 REM *                              *
140 REM ********************************
150 REM *                              *
160 REM *     (SEHR EINFACHES MENUE)   *
170 REM *                              *
180 REM ********************************
190 :
200 L=1: G=1: A=1: REM LEVEL, GESCHWINDIGKEI
    T UND ANZAHL DER SPIELER VORBELEGEN
210 COLOR 0,1: COLOR 4,1: PRINT CHR$(5): SCN
    CLR
220 PRINT TAB(7)"BEISPIELMENUE FUER C16/116"
230 PRINT TAB(7)"------------------------"
240 PRINT "{4DOWN}"
250 PRINT TAB(3);CHR$(18)" 1 " CHR$(146);TAB
    (10);"SPIELSTUFE{DOWN}"
260 PRINT TAB(3);CHR$(18)" 2 " CHR$(146);TAB
    (10);"GESCHWINDIGKEIT{DOWN}"
270 PRINT TAB(3);CHR$(18)" 3 " CHR$(146);TAB
    (10);"ANZAHL DER SPIELER{DOWN}"
280 PRINT TAB(3);CHR$(18)" 4 " CHR$(146);TAB
    (10);"AUSWAHL BEENDEN{2DOWN}"
290 PRINT "SPIELSTUFE :",L
300 PRINT "GESCHWINDIGKEIT :",G
310 PRINT "ANZAHL DER SPIELER :"A
320 INPUT "{2DOWN}WELCHEN MENUEPUNKT";P
330 IF P<1 OR P>4 OR P<>INT(P) THEN 320: REM
     NUR EINGABEN 1,2,3,4 ZULASSEN
340 :
350 REM UNTERPROGRAMME ANSPRINGEN
360 ON P GOTO 380,450,520,590
370 :
380 REM UNTERPROGRAMM VON MENUEPUNKT 1
390 REM ------------------------------
400 :
410 INPUT "SPIELSTUFE (1,2,3)";L
420 IF L<1 OR L>3 OR L<>INT(L) THEN 410: REM
     NUR RICHTIGE EINGABEN ZULASSEN
430 GOTO 210
440 :
450 REM UNTERPROGRAMM VON MENUEPUNKT 2
460 REM ------------------------------
470 :
480 INPUT "GESCHWINDIGKEIT (1,2,3)";G
490 IF G<1 OR G>3 OR G<>INT(G) THEN 480: REM
     NUR RICHTIGE EINGABEN ZULASSEN
500 GOTO 210
510 :
520 REM UNTERPROGRAMM VON MENUEPUNKT 3
530 REM ------------------------------
540 :
550 INPUT "ANZAHL DER SPIELER (1,2)";A
560 IF A<>1 AND A<>2 THEN 550
570 GOTO 210
580 :
590 REM UNTERPROGRAMM VON MENUEPUNKT 4
600 REM ------------------------------
610 :
620 PRINT "{UP}AUSWAHL UEBER MENUEPUNKT 4 BE
    ENDET.": END
```

Listing 1. Sehr einfaches Menü mit ON n GOTO-Befehl

```
100 REM ********************************
110 REM *                              *
120 REM * B E I S P I E L - M E N U E *
130 REM *                              *
140 REM ********************************
150 REM *                              *
160 REM *     (MIT WINDOW-TECHNIK)     *
170 REM *                              *
180 REM ********************************
190 :
200 L=1: G=1: A=1: REM LEVEL, GESCHWINDIGKEI
    T UND ANZAHL DER SPIELER VORBELEGEN
210 COLOR 0,1: COLOR 4,1: PRINT CHR$(5): SCN
    CLR
220 PRINT TAB(7)"BEISPIELMENUE FUER C16/116"
230 PRINT TAB(7)"------------------------"
240 PRINT "{5DOWN}"
250 PRINT TAB(3);CHR$(18)" 1 " CHR$(146);TAB
    (10);"SPIELSTUFE{DOWN}"
260 PRINT TAB(3);CHR$(18)" 2 " CHR$(146);TAB
    (10);"GESCHWINDIGKEIT{DOWN}"
270 PRINT TAB(3);CHR$(18)" 3 " CHR$(146);TAB
    (10);"ANZAHL DER SPIELER{DOWN}"
280 PRINT TAB(3);CHR$(18)" 4 " CHR$(146);TAB
    (10);"AUSWAHL BEENDEN{2DOWN}"
290 REM NACH DEN NORMALEN PRINT-ZEILEN  WERD
    EN JETZT DIE WINDOWS GEDRUCKT
300 :
310 REM ZUERST DAS COPYRIGHT-WINDOW:
320 REM ============================
330 :
340 CHAR 1,5,2,"U**********I"
350 CHAR 1,5,3,"⎕COPYRIGHT⎕"
360 CHAR 1,5,4,"⎕ WINDOW: ⎕"
370 CHAR 1,5,5,"@**********W"
380 CHAR 1,5,6,"⎕(C) 64'ER⎕"
390 CHAR 1,5,7,"J**********K"
400 :
410 REM NUN DAS LEVEL-WINDOW:
420 REM =====================
430 CHAR 1,22,5,"{RVSON,15SPACE}"
440 CHAR 1,22,6,"{RVSON} SPIELSTUFE:"+STR$(L
    )+" "
450 CHAR 1,22,7,"{RVSON,15SPACE,RVOFF}"
460 :
470 REM UND DAS GESCHWINDIGKEITSWINDOW:
480 REM ===============================
490 :
500 CHAR 1,28,10,"F**********S"
510 CHAR 1,28,11,"⎕TEMPO:"+STR$(G)+" ⎕"
520 CHAR 1,28,12,"Z**********X"
530 :
540 REM UND DAS SPIELERZAHL-WINDOW:
550 REM ===========================
560 :
570 CHAR 1,25,16,"**************"
580 CHAR 1,25,17,"*"+STR$(A)+" SPIELER *"
590 CHAR 1,25,18,"**************{DOWN}"+CHR$(
    13)
600 :
610 :
620 PRINT "BITTE ENTSPRECHENDE TASTE DRUECKE
    N !"
630 :
640 GET KEY A$
650 IF A$ = "1" THEN L=L+1: IF L>3 THEN L=1
660 IF A$ = "2" THEN G=G+1: IF G>3 THEN G=1
670 IF A$ = "3" THEN A=A+1: IF A>2 THEN A=1
680 IF A$ = "4" THEN END
690 :
700 GOTO 290
```

Listing 2. Listing mit Window-Technik, einfach gelöst

```
100 REM ********************************
110 REM *                              *
120 REM * M E N U E S T E U E R U N G *
130 REM *                              *
140 REM ********************************
150 REM *                              *
160 REM * VON FLORIAN MUELLER 10/1985 *
170 REM *                              *
180 REM ********************************
190 :
200 REM ***      INITIALISIERUNG      ***
210 REM ***      ===============      ***
220 :
230 CLR
240 A = 5: REM FUENF MENUEPUNKTE
250 DIM R$ (A-1): REM REVERS-TABELLE DIMENSI
    ONIEREN
260 P = 0: REM MENUEPUNKT 0 EINSTELLEN
270 R$ (P) = CHR$(18): REM MENUEPUNKT 0 REVE
    RS
280 COLOR 0,1: COLOR 4,1: PRINT CHR$(142)
290 PRINT CHR$(147): REM BILDSCHIRM LOESCHEN
300 :
310 REM *** AUSGABE DER MENUEPUNKTE ***
320 REM *** ======================== ***
330 :
340 PRINT CHR$(19);: REM CURSOR HOME
350 PRINT "{YELLOW,3SPACE}MENUESTEUERUNG - B
    EISPIELPROGRAMM"
360 PRINT "{3SPACE}=========================
    ========"
370 PRINT : PRINT : PRINT
380 PRINT TAB(9);R$(0);"{RED}MENUEPUNKT A{RV
    OFF} (0){DOWN}"
390 PRINT TAB(9);R$(1);"{CYAN}MENUEPUNKT B{R
    VOFF} (1){DOWN}"
400 PRINT TAB(9);R$(2);"{PURPLE}MENUEPUNKT C
    {RVOFF} (2){DOWN}"
410 PRINT TAB(9);R$(3);"{GREEN}MENUEPUNKT D{
    RVOFF} (3){DOWN}"
420 PRINT TAB(9);R$(4);"{BLUE}MENUEPUNKT E{R
    VOFF} (4){DOWN}"
430 PRINT : PRINT "{GREY3}DERZEIT ANGEWAEHLT
    ER MENUEPUNKT:"P
440 :
450 REM ***      TASTATURABFRAGE      ***
460 REM ***      ===============      ***
470 :
480 GET T$: IF T$="" THEN 480
490 IF T$="{F1}" THEN ON P+1 GOTO 610,680,73
    0,780,830
500 IF T$="{F7}" THEN PRINT "{2DOWN}PROGRAMM
    ABBRUCH.": END
510 R$(P) = CHR$(0)
520 IF T$="{DOWN}" THEN P = P+1 + A*(P=(A-1)
    )
530 IF T$="{UP}" THEN P = P-1 - A*(P=0)
540 IF T$="{HOME}" THEN P = 0
550 R$(P) = CHR$(18)
560 GOTO 310
570 :
580 REM *** MENUEPUNKTE  (ROUTINEN) ***
590 REM *** ======================== ***
600 :
610 REM MENUEPUNKT 1
620 PRINT "{CLR}SIE HABEN DEN OBERSTEN MENUE
    PUNKT ANGE-"
630 PRINT "{DOWN}WAEHLT. DIESER HAT IN DER I
    NTERNEN NU-"
640 PRINT "{DOWN}MERIERUNG DES PROGRAMMS DIE
     NUMMER:"P
650 PRINT "{2DOWN}BITTE DRUECKEN SIE EINE TA
    STE ";
660 POKE 204,0: WAIT 198,1: POKE 204,1: POKE
     198,0: GOTO 290
670 :
680 REM MENUEPUNKT 2
690 PRINT "{CLR}SIE HABEN DEN ZWEITOBERSTEN
    MENUEPUNKT"
700 PRINT "{DOWN}ANGEWAEHLT. INTERNE NUMMER:
     "P
710 GOTO 650
720 :
730 REM MENUEPUNKT 3
740 PRINT "{CLR}SIE HABEN DEN MITTLEREN MENU
    EPUNKT ANGE-"
750 PRINT "WAEHLT. DESSEN INTERNE NUMMER IST
    :"P
760 GOTO 650
770 :
780 REM MENUEPUNKT 4
790 PRINT "{CLR}SIE HABEN DEN ZWEITUNTERSTEN
     MENUEPUNKT"
800 PRINT "{DOWN}ANGEWAEHLT. INTERNE NUMMER:
    "P
810 GOTO 650
820 :
830 REM MENUEPUNKT 5
840 PRINT "{CLR}SIE HABEN DEN UNTERSTEN MENU
    EPUNKT ANGE-"
850 PRINT "WAEHLT. INTERNE NUMMER:"P
860 GOTO 650
```

Listing 3. Vertikal orientiertes Menü.
Vor dem Start unbedingt im Direktmodus eingeben:
KEY 1,CHR$(133):KEY 7,CHR$(136)

```
100 REM ********************************
110 REM *                              *
120 REM * M E N U E S T E U E R U N G *
130 REM *                              *
140 REM ****** GEKUERZTE VERSION ******
150 REM *                              *
160 REM * VON FLORIAN MUELLER 10/1985 *
170 REM *                              *
180 REM ********************************
190 :
200 REM INITIALISIERUNG
210 CLR : DIM R$(4): P=0: R$(P)="{RVSON}"
220 COLOR 0,1: COLOR 4,1: PRINT "{CLR}" CHR$
    (142)
230 REM AUSGABE DER MENUEPUNKTE
240 PRINT "{HOME,YELLOW,3SPACE}MENUESTEUERUN
    G - BEISPIELPROGRAMM"
250 PRINT "{3SPACE}=========================
    ========{2DOWN}"
260 PRINT TAB(11);R$(0);"{RED}MENUEPUNKT A{D
    OWN}"
270 PRINT TAB(11);R$(1);"{CYAN}MENUEPUNKT B{
    DOWN}"
280 PRINT TAB(11);R$(2);"{PURPLE}MENUEPUNKT
    C{DOWN}"
290 PRINT TAB(11);R$(3);"{GREEN}MENUEPUNKT D
    {DOWN}"
300 PRINT TAB(11);R$(4);"{BLUE}MENUEPUNKT E{
    DOWN,GREY3}"
310 REM TASTATURABFRAGE
320 GET T$: IF T$="" THEN 320
330 IF T$="{F1}" THEN ON P+1 GOTO 420,470,50
    0,530,560
340 IF T$="{F7}" THEN PRINT "{2DOWN}PROGRAMM
    ABBRUCH.": END
350 R$(P)=""
360 IF T$="{DOWN}" THEN P=P+1+5*(P=4)
370 IF T$="{UP}" THEN P=P-1-5*(P=0)
380 IF T$="{HOME}" THEN P=0
390 R$(P)="{RVSON}"
400 GOTO 230
410 REM MENUEPUNKTE (ROUTINEN)
420 PRINT "{CLR}SIE HABEN DEN OBERSTEN MENUE
    PUNKT ANGE-"
```

```
430 PRINT "{DOWN}WAEHLT. DIESER HAT IN DER I
    NTERNEN NU-"
440 PRINT "{DOWN}MERIERUNG DES PROGRAMMS DIE
     NUMMER:"P
450 PRINT "{2DOWN}BITTE DRUECKEN SIE EINE TA
    STE ";
460 POKE 204,0: WAIT 198,1: POKE 204,1: POKE
     198,0: GOTO 220
470 PRINT "{CLR}SIE HABEN DEN ZWEITOBERSTEN
    MENUEPUNKT"
480 PRINT "{DOWN}ANGEWAEHLT. INTERNE NUMMER:
     "P
490 GOTO 450
500 PRINT "{CLR}SIE HABEN DEN MITTLEREN MENU
    EPUNKT ANGE-"
510 PRINT "WAEHLT. DESSEN INTERNE NUMMER IST
    :"P
520 GOTO 450
530 PRINT "{CLR}SIE HABEN DEN ZWEITUNTERSTEN
     MENUEPUNKT"
540 PRINT "{DOWN}ANGEWAEHLT. INTERNE NUMMER:
     "P
550 GOTO 450
560 PRINT "{CLR}SIE HABEN DEN UNTERSTEN MENU
    EPUNKT ANGE-"
570 PRINT "WAEHLT. INTERNE NUMMER:"P
580 GOTO 450
```

Listing 4. Verkürztes vertikales Menü.
Vor dem Start unbedingt im Direktmodus eingeben:
KEY 1,CHR$(133):KEY 7,CHR$(136)

```
100 REM ********************************
110 REM *                              *
120 REM * M E N U E S T E U E R U N G *
130 REM *                              *
140 REM ********* HORIZONTAL **********
150 REM *                              *
160 REM * VON FLORIAN MUELLER 10/1985 *
170 REM *                              *
180 REM ********************************
190 :
200 REM INITIALISIERUNG
210 CLR : DIM R$(4): P=0: R$(P)="{RVSON}"
220 COLOR 0,1: COLOR 4,1: PRINT "{CLR}" CHR$
    (142)
230 REM AUSGABE DER MENUEPUNKTE
240 PRINT "{HOME,YELLOW,3SPACE}MENUESTEUERUN
    G - BEISPIELPROGRAMM"
250 PRINT "{3SPACE}========================
    ========{2DOWN}"
260 PRINT "{DOWN}DIESMAL SIND DIE MENUEPUNKT
    E HORIZONTAL"
270 PRINT "{DOWN}ANGEORDNET.{9DOWN}"
280 PRINT R$(0)"{RED}PUNKT A{RVOFF} "R$(1)"{
    CYAN}PUNKT B{RVOFF} "R$(2)"{PURPLE}PUNKT
     C{RVOFF} ";
290 PRINT R$(3)"{GREEN}PUNKT D{RVOFF} "R$(4)
    "{BLUE}PUNKT E{GREY3}"
300 :
310 REM TASTATURABFRAGE
320 GET T$: IF T$="" THEN 320
330 IF T$="{F1}" THEN PRINT "{DOWN}ANGEWAEHL
    TER PUNKT:"P: END
340 IF T$="{F7}" THEN PRINT "{2DOWN}PROGRAMM
    ABBRUCH.": END
350 R$(P)=""
360 IF T$="{RIGHT}" THEN P=P+1+5*(P=4)
370 IF T$="{LEFT}" THEN P=P-1-5*(P=0)
380 IF T$="{HOME}" THEN P=0
390 R$(P)="{RVSON}"
400 GOTO 230
```

Listing 5. Horizontal orientiertes Menü.
Vor dem Start unbedingt im Direktmodus eingeben:
KEY 1,CHR$(133):KEY 7,CHR$(136)

```
100 REM ********************************
110 REM *                              *
120 REM * M E N U E S T E U E R U N G *
130 REM *                              *
140 REM ***** HORIZONTAL/VERTIKAL *****
150 REM *                              *
160 REM * VON FLORIAN MUELLER 10/1985 *
170 REM *                              *
180 REM ********************************
190 :
200 REM INITIALISIERUNG
210 CLR : DIM R$(5): P=0: R$(P)="{RVSON}"
220 COLOR 0,1: COLOR 4,1: PRINT "{CLR}" CHR$
    (142)
230 REM AUSGABE DER MENUEPUNKTE
240 PRINT "{HOME,YELLOW,3SPACE}MENUESTEUERUN
    G - BEISPIELPROGRAMM"
250 PRINT "{3SPACE}========================
    ========{2DOWN}"
260 PRINT "{DOWN}MENUEPUNKTE SIND HORIZONTAL
     & VERTIKAL"
270 PRINT "{DOWN}ANGEORDNET.{5DOWN}"
280 PRINT TAB(10)R$(0)"{RED}PUNKT A{RVOFF,6S
    PACE}"R$(3)"{GREEN}PUNKT D{DOWN}"
290 PRINT TAB(10)R$(1)"{CYAN}PUNKT B{RVOFF,6
    SPACE}"R$(4)"{BLUE}PUNKT E{DOWN}"
300 PRINT TAB(10)R$(2)"{PURPLE}PUNKT C{RVOFF
    ,6SPACE}"R$(5)"{ORANGE}PUNKT F{GREY3}"
310 REM TASTATURABFRAGE
320 GET T$: IF T$="" THEN 320
330 IF T$="{F1}" THEN PRINT "{DOWN}ANGEWAEHL
    TER PUNKT:"P: END
340 IF T$="{F7}" THEN PRINT "{2DOWN}PROGRAMM
    ABBRUCH.": END
350 R$(P)=""
360 IF T$="{DOWN}" THEN P=P+1+3*(P=2 OR P=5)
370 IF T$="{UP}" THEN P=P-1-3*(P=0 OR P=3)
380 IF T$="{RIGHT}" THEN P=P+3+6*(P>2)
390 IF T$="{LEFT}" THEN P=P-3-6*(P<3)
400 IF T$="{HOME}" THEN P=0
410 R$(P)="{RVSON}"
420 GOTO 230
```

Listing 6. Zweidimensionales Menü.
Vor dem Start unbedingt im Direktmodus eingeben:
KEY 1,CHR$(133):KEY 7,CHR$(136)

```
100 REM ********************************
110 REM *                              *
120 REM *     SICHERHEITSABFRAGEN      *
130 REM *                              *
140 REM ********************************
150 REM *                              *
160 REM * FLORIAN MUELLER      M-1985 *
170 REM *                              *
180 REM ********************************
190 :
200 CLR : DIM R$(1): P=0: R$(P)="{RVSON}"
210 PRINT "{CLR}SICHERHEITSABFRAGEN (J/N-ENT
    SCHEIDUNGEN)"
220 PRINT "{HOME,6DOWN}WOLLEN SIE EINE ERKLA
    ERUNG? "R$(0)"JA{RVOFF}/"R$(1)"NEIN"
230 GET T$: IF T$="" THEN 230
240 R$(P)=""
250 IF T$="{RIGHT}" OR T$="{LEFT}" OR T$=" "
     THEN P=P+1+2*(P=1)
260 IF T$<>"{F1}" AND T$<>CHR$(13) THEN R$(P
    )="{RVSON}": GOTO 220
270 PRINT "{2DOWN,RVSON}ERGEBNIS:"P"{DOWN}"
280 IF P=0 THEN PRINT "ERKLAERUNG: SIEHE ART
    IKEL"
290 IF P=1 THEN PRINT "DANN EBEN NICHT!"
```

Listing 7. Ja/Nein-Abfrage. Vor dem Start unbedingt im Direktmodus eingeben: KEY 1,CHR$(133):KEY 7,CHR$(136)