

Fortsetzung von Seite 158

stellen) aufgeteilt. Das linke Halbbyte (4 Bit) einer Speicherstelle trägt die Informationen über senkrechte, das rechte Halbbyte über waagerechte Start/Endpositionen. Das 1. Bit im Halbbyte wird für Endpunkte gesetzt, das 3. Bit für Startpunkte. Diese Informationen werden später bei der Ausgabe des Rätsels auf dem Drucker benötigt, um an den richtigen Stellen die Nummer der zugehörigen Fragestellung einzutragen zu können.

ne braucht dann nur den Paß des Wortes zu überprüfen und bei korrekter Eintragung ins Basic zurückzukehren. Paßt das Wort allerdings nicht, verfällt die Routine in den Automodus. Das Maschinenprogramm durchläuft im Automodus eine Schleife, die zunächst durch Auslesen des Rauschgenerators im Soundchip ein Wort aus dem Wortschatz per Zufall bestimmt. Dann erfolgt auf die gleiche Weise die Auswahl eines Startpunktes

Hinweise zum Eintippen

In diesem Listing wurden die meisten Steuerzeichen umgesetzt in Buchstabenkombinationen, die in eckigen Klammern stehen. Es bedeuten:

clr = Shift Clear/Home-Taste	rvof = Revers off
home = Home-Taste	crsr = Cursor rechts
cyan = Control und 4	crsl = Cursor links
whit = Control und 2	(crsl)*28 = 28 mal
red = Control und 3	Cursor links)
rvon = Revers on	crsd = Cursor unten

In Zeile 51 bis 56 (Listing 3) bedeuten die Grafikzeichen (von oben nach unten) f7,f4,f1,f6,f8,f2.

Ein reverses »Z« bedeutet die Farbe Hellblau = Commodore-Taste und 7.

b) Eintragungen im »Senkrecht«-Speicher (50576 bis 50975); in dieser Speicher-matrix werden die Ordnungsnummern der Fragestellungen für senkrechte Rätselwörter abgelegt. Bei der Druckausgabe wird diese Matrix abgefragt (Zeile 773 bis 777), um die Nummer der Fragestellung im entsprechenden Startfeld einzusetzen.

c) Eintragung im »Waagrecht«-Speicher (50976 bis 51375), wie unter b) Zeile 796 bis 800).

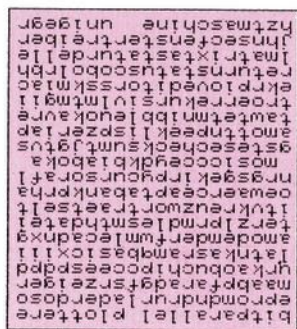
Um die Speicherinhalte vor dem Start des Rätselprogramms zu löschen, springt man die Routine mit »sys 50016« an (Zeile 937).

Nun zur Arbeitsweise der Routine während der automatischen Rätselerzeugung. Zunächst muß das Maschinenprogramm erst einmal wissen, ob es wegen einer Handeintragung angesprungen wurde oder zur automatischen Wortsuche. Dazu liest die Routine die Speicherstelle 26 aus. Ist das Ergebnis Null, wird in den Automodus verzweigt. Andernfalls wird in 26 die Länge des von Hand eingetragenen Wortes übergeben (siehe Zeile 662 bis 663). Die Routi-

ne im Bildspeicher. Ist dieser Punkt nicht geeignet für eine Eintragung, wird Zeile für Zeile des Wortfeldes nach einer Alternative gesucht. Im Falle eines Treffers startet der Wortvergleich. Das gewählte Wort wird mit dem Inhalt des Bildspeichers auf Übereinstimmung geprüft. Paßt es nicht, kommt das nächste Wort aus dem Wortschatz-Array an die Reihe. Bei Erfolg kehrt »Such« ins Basic zurück, wenn nicht, beginnt die Schleife von Neuem.

Damit der Zugriff auf das Array klappt, muß es nur als erstes im Basic-Programm definiert worden sein (Zeile 925). »Primitiv« werden Sie vielleicht anmerken. Richtig, aber Computer sind nun mal (sehr) schnelle Idioten.

(Gert Büttgenbach/gk)



Lösung des Kreuzworträtsels

Programmierwettbewerb:

Dokumentationshilfe

Insgesamt 1000 Mark zu gewinnen. Möchte man ein Programm analysieren oder schreiben, und die Dokumentation ist nicht oder nur mangelhaft vorhanden, ist eine automatische Dokumentationshilfe ein interessantes Werkzeug.

Die Aufgabe, die wir diesmal stellen, ist nicht nur eine Herausforderung an Programmierer, sondern soll zudem für Software-Entwickler ein nützliches Utility sein. Es geht um die Programmierung einer erweiterten Crossreferenzliste. Eine Crossreferenzliste durchsucht per Definition ein beliebiges Programm nach Variablen und Sprungbefehlen und gibt sie auf einem Drucker in gut lesbarer Form aus. Wir wollen aber in diesem Programmierwettbewerb ein vollständiges Werkzeug zur Dokumentation eines sich in der Entwicklung befindlichen oder fertigen Programms erhalten. Im einzelnen sollte das Programm folgendes können:

1. Alle Programmzeilennummern drucken, die Sprünge enthalten. Ausgegeben werden soll die Zeilennummer, dahinter die Zeilen, die angesprungen werden.

2. Ausgabe aller Programmzeilen, die angesprungen werden, wenn möglich mit den Zeilen, von denen aus der Sprung erfolgt.

3. Ausgabe aller im Programm verwendeten Variablen.

3.1 In der Reihenfolge, wie sie im Programm auftauchen.

3.2 In sortierter Reihenfolge: Sortiert nach Gruppe (Integer, Real, Strings und Felder) sowie alphabetisch.

3.3 In welcher Zeile sie definiert werden (Variable =) und in welcher Zeile sie benutzt werden (= Variable).

3.4 Es soll zu jeder Variable ein Kommentar eingegeben werden können.

4. Denkbar wäre auch, die ganze Prozedur innerhalb wählbarer Grenzen (zum Beispiel zwischen Zeile 1000 und 2000) eines Programms ablaufen zu lassen.

Wie Sie aus dem letzten Punkt ersehen können, sind den Ideen keine Grenzen gesetzt. Wichtig ist vor allen Dingen, daß ein komplettes Dokumentationsprogramm für die eigene Entwicklung und zur Analyse fremder Programme zustande kommt.

So könnte eine automatische Aufschlüsselung nach Zeilennummern oder die Erstellung eines Fluß- oder Nassi-Shneidermann-Diagramms durchaus mit eingebaut werden. Lassen Sie Ihre Phantasie spielen und dokumentieren eigene und fremde Programme auf die bestmögliche Art und Weise.

Es wird mindestens zwei Gewinner geben: Einer für die beste Lösung in Basic, der andere für das beste Assembler-Programm.

Wenn Ihre Lösung von der oben genannten Aufgabenstellung etwas abweicht, so ist das keine Disqualifikation. Bewertungskriterien werden vor allem sein: Nutzbarkeit, Übersichtlichkeit, Schnelligkeit und Komfort.

Schicken Sie Ihre Lösung unter dem Stichwort

»Programmierwettbewerb: Dokumentationshilfe« an folgende Adresse:

Markt & Technik Verlag AG, Redaktion 64'er, Hans-Pinsel-Str. 2, 8013 Haar bei München