

Einsteigen - Verstehen - Beherrschen

DM 3,80 öS 30 sfr 3,80

computer kurs

Computerfilme

Das Grafiktablett

PEEK und POKE

Speichertechnik im Umbruch

Gesetze der Logik

Dreidimensionale Spiele

Heft

9

Ein wöchentliches Sammelwerk

Programmierkurse
BASIC und LOGO

computer kurs

Heft 9

Inhalt

Computer Welt

Computerfilme 225

Zeichentricks per Rechner

3-D-Computerspiele 246

Hardware

Speichertechnik im Umbruch 228

Von der Elektronenröhre zum Biospeicher

Sord M5 230

Fragen und Antworten

Welche Sprache verstehen Computer? 233

Software

Geordnete Dateien 234

Elektronische Namensregister

Peripherie

Das Grafiktablett 238

Tips für die Praxis

Wir stellen vor: Sound und Grafik 240

BASIC

Dem puren Zufall ausgeliefert 242

LOGO

Farb-„DEMO“ und LOGO-Telefonbuch 248

Bits und Bytes

Gesetze des Denkens 250

George Booles Regeln der Mathematik

PEEK und POKE 252

Zwei wichtige BASIC-Programmierbefehle

Fachwörter auf einen Blick

WIE SIE JEDE WOCHE IHR HEFT BEKOMMEN

Computer Kurs ist ein wöchentlich erscheinendes Sammelwerk. Die Gesamtzahl der Hefte ergibt ein vollständiges Computer-Nachschlagewerk. Damit Sie jede Woche Ihr Heft erhalten, bitten Sie Ihren Zeitschriftenhändler, Computer Kurs für Sie zu reservieren.

Zurückliegende Hefte

Ihr Zeitschriftenhändler besorgt Ihnen gerne zurückliegende Hefte. Sie können sie aber auch direkt beim Verlag bestellen.

Deutschland: Das einzelne Heft kostet DM 3,80. Bitte füllen Sie eine Postzahlkarte aus an: Marshall Cavendish Int. Ltd. (MCI), Sammelwerk-Service, Postgiroamt Hamburg 48064-202, Postfach 105703, 2000 Hamburg 1, Kennwort: Computer Kurs

Österreich: Das einzelne Heft kostet öS 30. Bitte füllen Sie eine Zahlkarte aus an: Computer Kurs, Wollzeile 11, 1011 Wien, Postscheckkonto Wien 7857201 oder legen Sie Ihrer Bestellung einen Verrechnungsscheck bei. Kennwort: Computer Kurs.

Schweiz: Das einzelne Heft kostet sfr 3,80. Bitte überweisen Sie den Betrag durch die Post (grüner Einzahlungsschein) auf das Konto: Schmidt Agence AG, Kontonummer Basel 40-879, Kennwort: Computer Kurs, und notieren Sie ihre Bestellung auf der Rückseite des Giroabschnittes (rechter Abschnitt).

Abonnement

Sie können Computer Kurs auch alle 2 Wochen (je 2 Ausgaben) per Post zum gleichen Preis im Abonnement beziehen. Der Abopreis für 12 Ausgaben beträgt DM 45,60 inkl. MwSt., den wir Ihnen nach Eingang der Bestellung berechnen. Bitte senden Sie Ihre Bestellung an: Marshall Cavendish Int. Ltd. (MCI), Sammelwerk Service, Postgiroamt Hamburg 86853-201, Postfach 105703, 2000 Hamburg 1, Kennwort: Abo Computer Kurs. Bitte geben Sie an, ab welcher Nummer das Abo beginnen soll und ob Sie regelmäßig für jeweils 12 Folgen einen Sammelordner wünschen. Bei Bestellungen aus Österreich oder Schweiz senden Sie Ihren Auftrag bitte auch an die Hamburger Adresse. Berechnung und Zahlung erfolgen in Landeswährung zum Ladenpreis.

WICHTIG: Bei Ihren Bestellungen muß der linke Abschnitt der Zahlkarte Ihre vollständige Adresse enthalten, damit Sie die Hefte schnell und sicher erhalten. Überweisen Sie durch Ihre Bank, so muß die Überweisungskopie Ihre vollständige Anschrift gut leserlich enthalten.

SAMMELORDNER

Sie können die Sammelordner entweder direkt bei Ihrem Zeitschriftenhändler kaufen (falls nicht vorrätig, bestellt er sie gerne für Sie) oder aber Sie bestellen die Sammelordner für den gleichen Preis beim Verlag wie folgt.

Deutschland: Der Sammelordner kostet DM 12. Bitte füllen Sie eine Zahlkarte aus an: Marshall Cavendish International Ltd. (MCI), Sammelwerk-Service, Postgiroamt Hamburg 48064-202, Postfach 105703, 2000 Hamburg 1, Kennwort: Sammelordner Computer Kurs

Österreich: Der Sammelordner kostet öS 98. Bitte füllen Sie eine Zahlkarte aus an: Computer Kurs Wollzeile 11, 1011 Wien, Postscheckkonto Wien 7857201 oder legen Sie Ihrer Bestellung einen Verrechnungsscheck bei. Kennwort: Sammelordner Computer Kurs

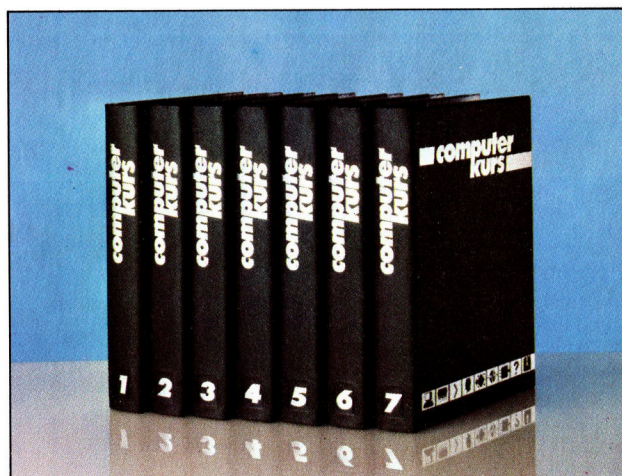
Schweiz: Der Sammelordner kostet sfr 15. Bitte überweisen Sie den Betrag durch die Post (grüner Einzahlungsschein) auf das Konto: Schmidt Agence AG, Kontonummer Basel 40-879, Kennwort: Sammelordner Computer Kurs, und notieren Sie Ihre Bestellung auf der Rückseite des Giroabschnittes (rechter Abschnitt).

INHALTSVERZEICHNIS

Alle 12 Hefte erscheint ein Teilindex. Die letzte Ausgabe von Computer Kurs enthält den Gesamtindex — darin einbezogen sind Kreuzverweise auf die Artikel, die mit dem gesuchten Stichwort in Verbindung stehen

Redaktion: Winfried Schmidt (verantw. f. d. Inhalt), Joachim Seidel, Elke Leibinger, Susanne Brandt, Sammelwerk Redaktions-Service GmbH, Paulstraße 3, 2000 Hamburg 1

Vertrieb: Marshall Cavendish International Ltd., Heidenkampsweg 74, 2000 Hamburg 1, Tel.: 040/23 40 80



© APSIF, Copenhagen, 1982, 1983; © Orbis Publishing Ltd., 1982, 1983; © Marshall Cavendish Ltd., 1984, 1985; **Druck:** E. Schwend GmbH, Schmollerstraße 31, 7170 Schwäbisch Hall



Computerfilme

Rechner eröffnen ungeahnte Dimensionen in der Trickfilmtechnik. Aufgrund der riesigen Speicherkapazität und der enormen Verarbeitungsgeschwindigkeit sind Grafiken nicht mehr von Fotografien zu unterscheiden.

Der Produktionsablauf bewegter Bilder, gleich ob beim Film oder beim Fernsehen, ist nur möglich durch die Unfähigkeit des Gehirns, einzelne Bilder „einzufrieren“. Dadurch, daß dem Auge eine rasche Bildfolge präsentiert wird, entsteht der Eindruck von tatsächlicher Bewegung.

Einer der ersten Versuche, die Illusion des bewegten Bildes zu erzeugen, bestand darin, daß man eine Trommel mit Schlitzen versah, auf ihrer Innenseite eine Abfolge von Zeichnungen befestigte und die Trommel drehte. Blickte man durch die Schlitze, sah man eine recht rohe Aufeinanderfolge dieser Bilder. Sehr bald ersetzten Fotos die Zeichnungen auf der Innenseite der Trommel. Voraussetzung für das nächste Entwicklungsstadium den Film, war das Vorhandensein relativ schnell reagierender fotografischer Materialien, die es ermöglichten, ein Bild in weniger als einer sechzehntel Sekunde aufzuzeichnen. Im Anfangsstadium der Filmindustrie wurden Filme mit einer Projektionsgeschwindigkeit von sechzehn Bildern pro Sekunde gezeigt.

24 Bilder pro Sekunde

Seltsamerweise dauerte es einige Zeit, bis die Idee des Zeichnens der Einzelbilder von Hand, das Fotografieren dieser Zeichnungen und schließlich die Projektion des Ergebnisses sich in der Filmindustrie durchsetzte. Bedenkt man, daß für jede Filmsekunde 24 Einzelbilder geschaffen werden müssen (das ist die Projektionsgeschwindigkeit von Filmen heute), leuchtet ein, daß die Produktion selbst eines fünfminütigen Films eine ungeheure Arbeit erfordert. In diesem Fall müssen 7200 einzelne Aufnahmen gemacht werden. So überrascht es nicht, daß diese Technik formalisiert wurde, da exakte Wiederholbarkeit unabdingbare Voraussetzung ist – Bugs Bunny muß von einer Sekunde zur nächsten eben immer noch aussehen wie Bugs Bunny.

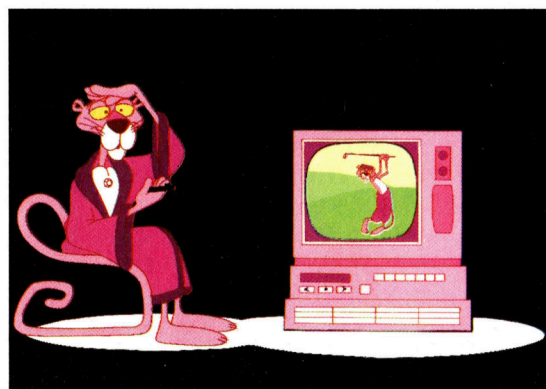
Aufgaben wie diese sind leichter und genauer von Maschinen durchzuführen. Wenn der Computer die Animation übernimmt – und dabei die Geschwindigkeit der Bewegung, den Wechsel von Perspektive und Gestalt, Licht und Schatten, Größe, Rhythmus und anderes mehr steuert –, kann sich der Künstler ganz auf die Qualität der Charaktere oder des

Gesamtbildes konzentrieren. In diesem Fall ist Animation keine wirklich grafische Kunst mehr, da der Künstler ausschließlich kreiert, die Bewegung aber durch den Computer erzeugt wird.

In der einfachsten Form benutzt man 'Sprites', um Charaktere zu schaffen, die dann auf den Bildschirm übertragen und bewegt werden, etwa so, wie man es aus simplen Videospiele kennt. Um die Illusion sowohl von



Beim herkömmlichen Zeichentrickfilm, wie beispielsweise „Der rosarote Panther“, muß jedes Bild einzeln gezeichnet werden, obwohl Grundelemente nur dann neu zu zeichnen sind, wenn sie ihre Form oder Position ändern. Die Arbeitsweise wird durch das Übereinanderlegen von Klarsichtfolien vereinfacht.





Abwechslung als auch von Bewegung (zum Beispiel beim Gehen) zu erzeugen, ist es erforderlich, daß ein Sprite ständig durch das nächste ersetzt wird. Wie bereits bekannt, ist die Erzeugung von Sprites relativ umständlich, gemessen an der grafischen Qualität der recht einfachen zweidimensionalen Ergebnisse.

Tiefe ins Bild

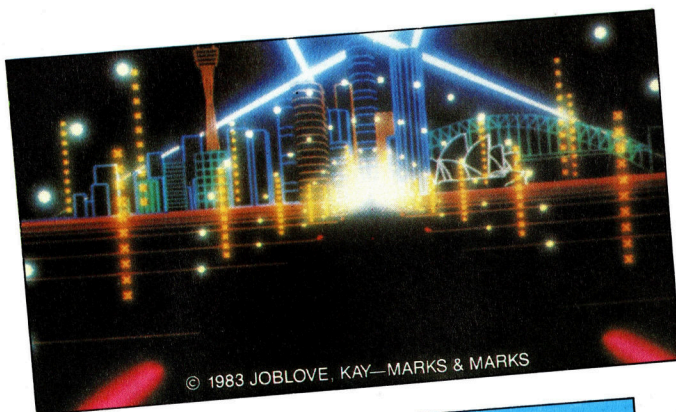
Anspruchsvollere Animation erfordert vom Programmierer die Entwicklung eines Algorithmus, der entsprechend den Gesetzen der Perspektive Tiefe ins Bild bringt. Objekte auf dem Bildschirm können dabei durch ihre X-, Y- und Z-Koordinaten definiert werden.

Wünschenswert sind ferner „runde“ Kurven. Eine Kurve wird durch drei Punkte bestimmt – ihre Endpunkte und den Scheitelpunkt (jener Punkt, der am weitesten von der gedachten Geraden der beiden Endpunkte entfernt liegt). Natürlich muß eine

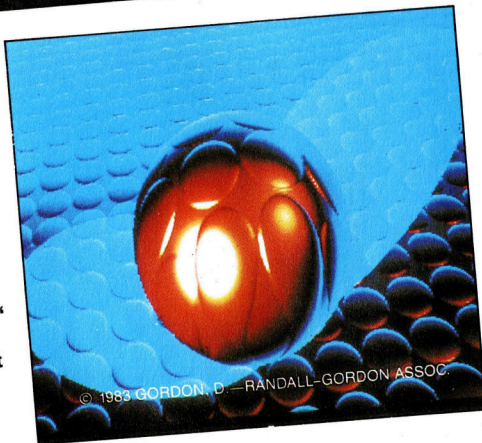
tierende Schatten hilft, die Umriss des Gegenstandes deutlicher zu machen. Höher entwickelte Software ermöglicht die Verwendung mehrerer Lichtquellen und erlaubt die Arbeit mit Reflexionen der Objekte.

Schatten und Farbe ergänzen sich. Selbst der einfachste Heimcomputer bietet mindestens acht Farbmöglichkeiten. Bei einem professionellen Grafikcomputer dagegen stehen 4096 Farben zur Verfügung. Das bedeutet, daß dieser Computer 16,7 Millionen Farbnuancen darstellen kann. Schatten- und Farbgebung sind in einem Befehl zusammengefaßt.

Beschäftigen wir uns nun mit der Simulation von Bewegung. Es ist relativ einfach, Bewegungen von geometrischen Objekten in Einzelphasen aufzulösen – selbst etwas so Komplexes wie die menschliche Hand. Das Ergebnis hängt natürlich von der Leistungsfähigkeit des Computers ab. Bedenken

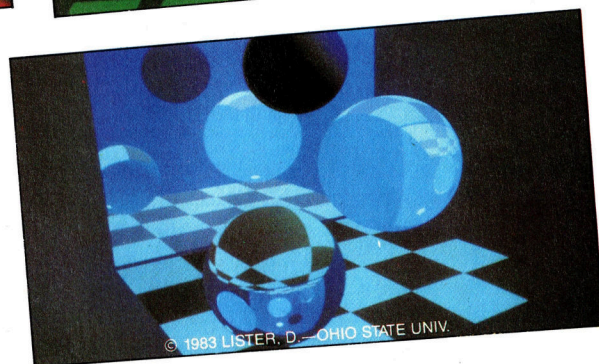
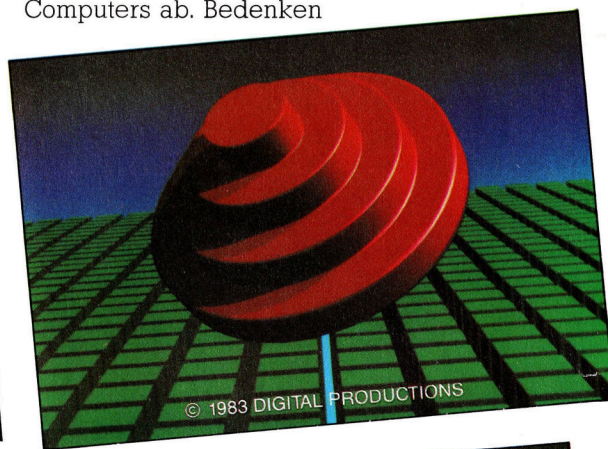


Bis vor kurzem wurden die Vorlagen für Film und Fernsehaufnahmen genauso erstellt wie bei Zeitschriften, das heißt, man fertigte entweder Dias oder Papierabzüge an. Bei 'Quantels Paint Box-System' dagegen wird darauf völlig verzichtet. Die „Bilder“ werden im Computer digitalisiert und direkt auf Videofilm aufgezeichnet.



kompliziertere Kurve, wie zum Beispiel ein 'S' in ihre einzelnen Bestandteile aufgelöst werden, um sie genau darstellen zu können. Dabei ist wichtig, daß das Programm dem Rechner verdeutlicht, daß hier eine Rundung und nicht eine Aneinanderreihung von Winkeln zu zeichnen ist.

Nun müssen Licht und Schatten in die Zeichnung gebracht werden. Dabei ist zunächst die genaue Festlegung der Position der Lichtquelle erforderlich. Der Teil des gezeichneten Objektes, der von der Lichtquelle bestrahlt wird, ist entsprechend hell. Der daraus resul-



Sie, daß zur Darstellung eines hochwertigen Bildes ein Monitor mit einer Auflösung von mindestens 1000 mal 1200 Pixels (Bildschirmpunkten) erforderlich ist. Für die Information, mit der Farbe und Helligkeit jedes dieser Pixels bestimmt werden, wird wenigstens ein Byte benötigt. Das bedeutet ein Mega-Byte pro Bildschirm. Die Erzeugung hochwertiger bewegter Bilder ist deshalb auf Heimcomputern unmöglich. Professionelle Trickfilmer bedienen sich der größten und leistungsfähigen Computer der Welt, was sich auch in ihren Honoraren niederschlägt.

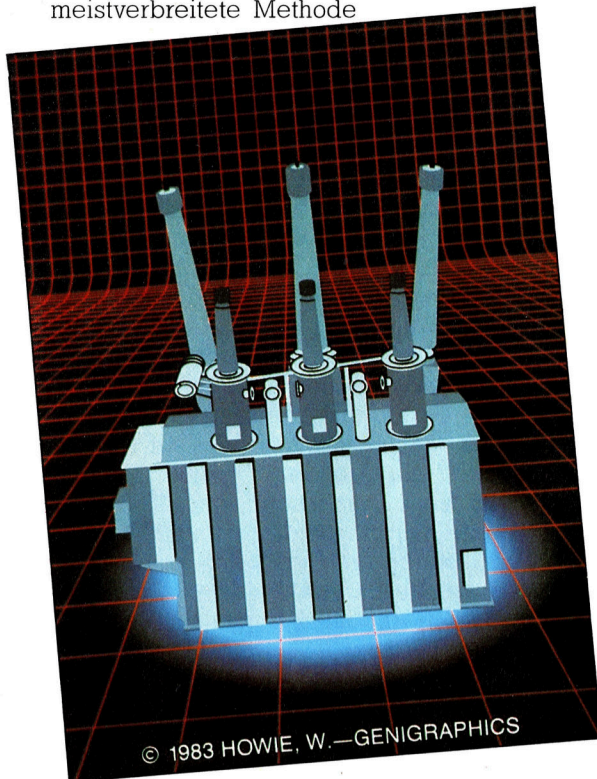
Verdeutlichen wir die Bewegung an einem so einfachen Objekt wie dem Würfel, der durch die Koordinaten seiner acht Eckpunkte definiert wird. Genau dieses Prinzip ist bei den



meisten anderen räumlichen Formen anwendbar. Der einzige Unterschied besteht in der zur Definition der Koordinaten erforderlichen Speicherkapazität. Entscheidend für die Echtzeit-Bewegung ist ferner die Verarbeitungsgeschwindigkeit des Prozessors. Je feiner die Zeichnungen definiert werden, desto mehr Speicherplatz ist erforderlich.

Die Auflösung eines normalen Fernsehbildschirms ist erheblich größer als die eines Spezialmonitors für Computer. Folglich wird jedes auf einem Monitor erzeugte Bild zumindest ebenso realistisch wie ein normales Fernsehbild sein.

Ein optimales Ergebnis ist aber nur mit hochentwickelter Hard- und Software möglich. Die meistverbreitete Methode



ist der sogenannte 'Bit-Pad-Digitiser', der einem großen, mit einem Drahtgeflecht versehenen Zeichenbrett vergleichbar ist. Die Position eines darauf geführten Stiftes wird vom Computer gelesen und auf dem Bildschirm angezeigt. Jede Art von Zeichentechnik ist dabei möglich. Die Darstellung des Zeichens auf dem Bildschirm kann vom Anwender ebenso definiert werden, als würde er einen Bleistift, eine Zeichenfeder oder den Pinsel verwenden. Ähnlich sind Farben durch Abruf einer Palette definierbar, einer Abfolge von Farben am Bildschirmrand, die der eines Malers vergleichbar ist.

Bewegung ins Bild

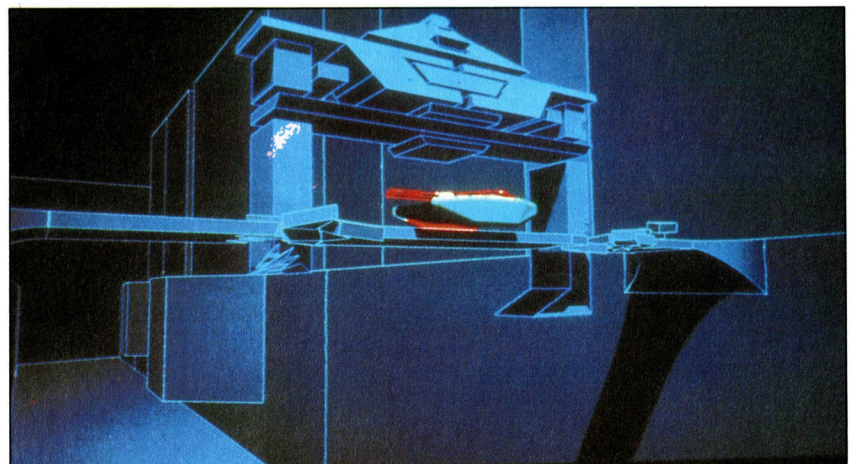
Farbmischung ist ebenso möglich wie Radieren, es können sogar „Zeichnungen“ übereinandergelegt werden. Wie aber kommt Bewegung ins Bild? Eine Möglichkeit besteht darin,



die herkömmliche Produktionsweise auf dem Computer nachzuvollziehen, indem man die Einzelbilder aneinanderreih, koloriert und die erstellten Rohsequenzen ablaufen läßt. So wird die Arbeitsweise zwar beschleunigt, jedoch läßt sich mit ausgefeilteren Programmen weit mehr anfangen. So wie eine Kurve mittels Programm gerundet wird, können ganze Szenen hergestellt werden, indem man lediglich das erste und letzte Bild einer Sequenz festlegt. Die dazwischenliegenden Zeichnungen, die man 'Phasenzeichnungen' nennt, werden in herkömmlichen Trickfilmstudios von dem sogenannten 'Inbetweener' (einem Assistenten des Designers) angefertigt. Eben diese Arbeit kann von einem Computer ausgeführt werden. Bisher bestand die Hauptarbeit bei der Trickfilmherstellung darin, daß die Illusion der Bewegung von Hand erfolgte, etwa durch einzelne Phasenzeichnungen oder vorgefertigte Folien; Darstellung von Bewegungen ist für den Computer geradezu maßgeschneidert: Wenn die Abläufe erst einmal festgelegt sind, bringt er das gewünschte Ergebnis.

Aufgrund der riesigen Speicherkapazität und Verarbeitungsgeschwindigkeit moderner Computer ist es möglich, Bilder zu schaffen, die von einem Foto nicht zu unterscheiden sind. Durch bestimmte Programmtechniken, die für statistische Problemlösungen entwickelt wurden, können die Bilder so modifiziert werden, daß man sie für real hält.

'TRON' war der erste Film, in dem Bilder durch Computer generiert wurden. Die Handlung spielt sowohl in der wirklichen Welt als auch im Inneren eines Computers. Die Realisierung dieser phantastischen Welt war nur auf dem Computer möglich.



Speichertechnik im Umbruch

Von der Elektronenröhre bis zum Biospeicher von morgen. Die Entwicklung zeigt: Datenspeicher werden immer kompakter und leistungsfähiger.

Jeder Computer muß in der Lage sein, einen gewissen Umfang an Informationen zu speichern. Selbst eine so einfache Operation wie das Addieren setzt voraus, daß mindestens eine Zahl (oder ein Digit) gespeichert wird, bis sie in der nächsten Stufe des Rechengangs gebraucht wird. Seit es Computer gibt, hat sich deren Fähigkeit, Informationen zu speichern, ständig verbessert – mit der Folge, daß immer umfangreichere oder vorher aufgrund geringerer Speicherkapazität unlösbare Probleme in Angriff genommen werden konnten.

Seit der Mensch die Elektrizität kennt, sucht er nach Wegen, sie zum Speichern von Informationen zu nutzen. Man kann sich Elektrizität auf verschiedene Weise bildhaft machen, als Elektronenfluß oder als schwingende Welle – das dominierende Merkmal ist immer die Bewegung. Da es unmöglich ist, etwas, das von Natur aus immer in Bewegung ist und sein muß, auf- oder zurückzuhalten, wurden indirekte Speichermethoden entwickelt. Eine Batterie zum Beispiel speichert Energie in chemi-

scher Form, ein Wasserkraftwerk setzt das im Wasser ruhende Energiepotential in den Turbinen frei. Die Suche nach Möglichkeiten, Daten in Form elektrischer Signale zu speichern, hat im Laufe der Zeit zu sehr unterschiedlichen Lösungen geführt.

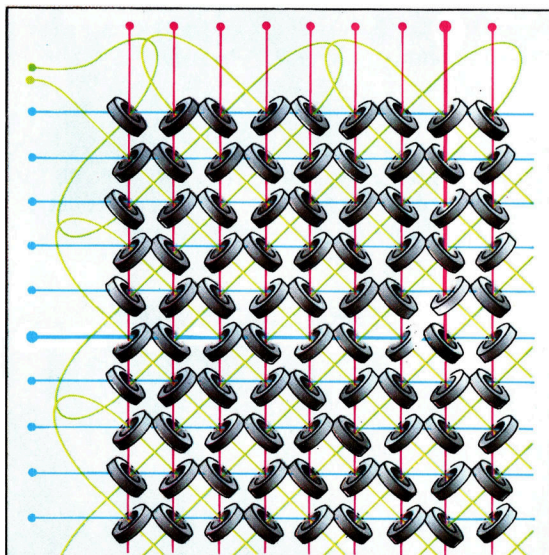
Eine der ersten brauchbaren Lösungen war die sogenannte Quecksilberverzögerungsleitung. Sie bestand aus einer mit Quecksilber gefüllten Glasröhre von einem Meter Länge und war an jedem Ende durch einen Quarzkristall abgeschlossen. Wurde an einem Ende ein elektrischer Impuls angelegt, geriet der Kristall in Schwingungen, die wiederum im Quecksilber eine Druckwelle erzeugten, die zum anderen Röhrenende wanderte. Dort traf die Druckwelle auf den Ausgangskristall, der dadurch zum Schwingen kam und ein elektrisches Signal als Ausgang lieferte. Da die Druckwelle zum Durchlaufen der Glasröhre eine gewisse Zeit benötigte (etwa 660 Mikrosekunden), stellte die Quecksilberöhre eine Verzögerungsstrecke dar, die zum vorübergehenden Speichern elektrischer Impulse genutzt werden konnte.

Verbesserte Speicherdauer

Um die Verzögerungszeit und damit die Speicherdauer zu verlängern, ließ man die Druckwelle in der Glasröhre mehrere Male hin und her laufen, was jedoch nicht endlos geschehen konnte, weil sich die Druckwelle abschwächte und dann verzerrte Signale verursachte. Immerhin gelang es auf diese Art und Weise, Speicherzeiten von der Dauer einiger Minuten zu realisieren.

Dieser Art der Datenspeicherung waren natürliche und technische Grenzen gesetzt. Die Speicherkapazität einer Einmeteröhre von 330 Bit (möglich durch eine Zeitschaltung zum Regulieren der Impulsfolge) konnte zwar durch Verlängern der Röhre erweitert werden, jedoch nur zum Preis einer verlängerten Zugriffszeit, denn jeder Impuls ließ sich ja erst am Ende der Verzögerungsstrecke erfassen.

Einen ganz anderen Weg ging in England F. C. Williams von der Universität Manchester. Er untersuchte die Möglichkeit, auf der Innenseite von Kathodenstrahlröhren (das sind die



Kernspeicher

Die Eisenringe auf einem Gitternetz können „adressiert“ werden, indem durch den entsprechenden horizontalen und vertikalen Draht Strom

geschickt wird. Der „lesende“ Draht (grün) erfasst Veränderungen im Magnetfeld des Rings und erkennt daraus, ob 1 oder 0 gespeichert ist.



Bildschirmröhren unserer heutigen Fernseher) erzeugte statische Ladungen als Speichermedium zu nutzen. Ein mit Hilfe einer Elektronenspritze auf der Innenseite aufgetragenes Muster aus statischen Ladungen kann durch ein Drahtgitter an der Außenseite des Bildschirms erfaßt werden. 1947 war man so weit: 2048 Bit konnten auf einem Schirm für die Dauer einiger Stunden gespeichert werden. Die Zugriffszeit war kurz, jedoch mußten die statischen Ladungen alle 30 Mikrosekunden „aufgefrischt“ werden, um ihr „Schwinden“ zu vermeiden.

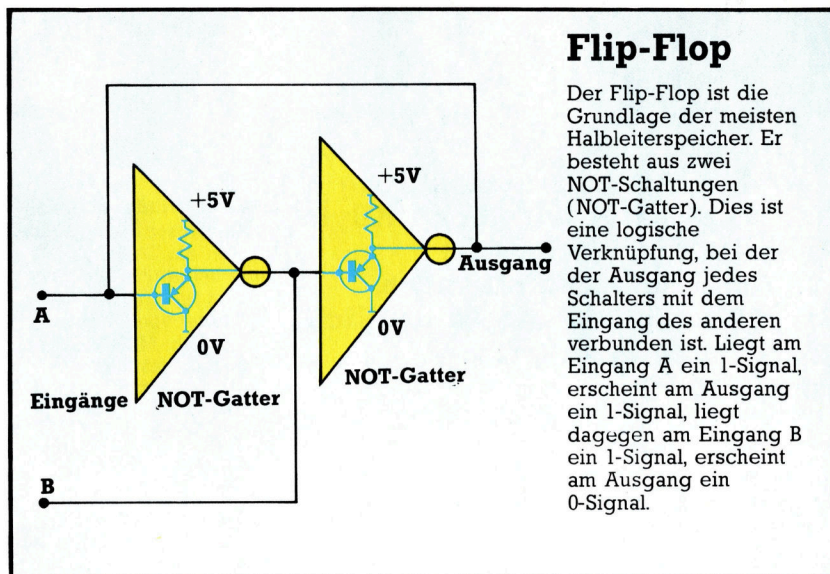
Erste Versuche mit Magnetbändern als Speichermedium liefen Ende 1940 in England in Verbindung mit LEO (Lyon's Electronic Office) und in den USA im Zusammenhang mit dem UNIVAC-Rechner (Universal Automatic Computer). Es war dies die erste Technologie, mit der sich große Informationsmengen billig und zuverlässig speichern ließen. Es gelang, acht Bit in einem Bandsektor unterzubringen, was bei 2360 Sektoren je Zentimeter und einer Bandlänge von 61 Metern (und darüber) permanente Speicher im Megabyte-Bereich realisierbar machte.

Der Flip-Flop

Mit der großmaßstäblichen Integration von Transistoren auf einem einzigen Chip wurde eine der ersten Methoden zur Datenspeicherung (in Verwendung seit ENIAC) neu belebt: Der Flip-Flop besteht aus zwei aneinanderliegenden elektronischen Schaltern, wobei der Ausgang eines jeden Schalters in den Eingang des anderen geführt ist. Auf diese Weise kann ein Impuls im Flip-Flop so lange „gefangen“ gehalten werden, bis er gebraucht wird.

„Statische“ RAM-Chips, wie sie in den früheren Heimcomputern zu finden sind, enthalten Tausende von bistabilen Stromkreisen – einen für jedes Bit. Die meisten neueren Computer dagegen sind mit „dynamischen“ RAMs ausgestattet, von denen jeder einen elektrischen Impuls speichern kann.

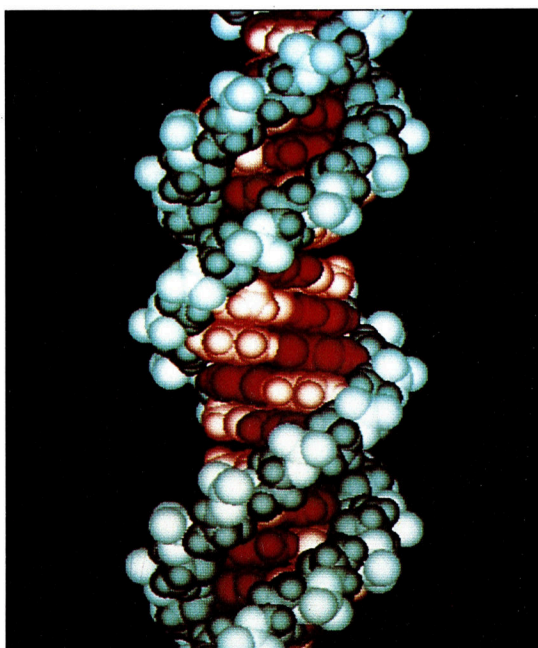
Zu den neuesten Entwicklungen auf dem Gebiet der Datenspeicher gehören die Magnetblasenspeicher (im Fachjargon 'Bubbles' genannt) und die optischen Laser-Speicher. Die Laser-Speicher sind den Platten ähnlich, wie sie zum Speichern von Musik oder Videofilmen verwendet werden, werden jedoch durch einen Laserstrahl abgetastet. Bei den Bubbles handelt es sich um winzige Blasen, die innerhalb eines Chips aus magnetischem Material erzeugt werden. Diese Blasen lassen sich in Schleifen bewegen; das Vorhanden- oder Nicht-Vorhandensein einer Blase stellt den Binärzustand 1 oder 0 dar. Um eine möglichst kurze Zugriffszeit zu erhalten, werden die Schleifen klein gehalten. Die Dichte der Blasen ist so groß, daß die Magnetblasenspeicher potentiell in der Lage sind, die derzeitigen auf dem Markt befindlichen Cassetten-



und Plattenspeicher vollwertig zu ersetzen.

Das Neueste, was an Speicherarten auf uns zukommt, sind kryogenische, d. h. kälteerzeugende Speicher. Bei Temperaturen nahe des absoluten Nullpunktes wird der elektrische Widerstand jedes Mediums nahezu Null, und es entsteht die sogenannte Supraleitfähigkeit. Damit wird es theoretisch möglich, einen elektrischen Impuls zu speichern, der demjenigen eines einzelnen Elektrons gleichkommt. Die Impulse, mit denen unsere heutigen Chips arbeiten, sind zwar sehr klein, kommen aber immer noch der Ladung von einigen Millionen Elektronen gleich.

Mit Hilfe der Supraleitfähigkeit, die durch Abkühlen des Speichermediums in flüssigem Helium erreicht wird, kann die Schnelligkeit und Kapazität von Speichern noch weit über das hinaus angehoben werden, was mit den heute bekannten Speicherarten möglich ist.



Neuerdings befassen sich Wissenschaftler mit Speichern, die auf Molekularebene arbeiten, d. h. mit einzelnen Elektronen. Dies ist aber mehr ein Gebiet der Biotechnik als der Microelektronik. Es ist beispielsweise bekannt, daß die Struktur eines DNA-Moleküls den genetischen Code speichert, der die Art und Weise unseres Wachstums steuert. Es werden jedoch noch sehr viele Jahre vergehen, bis es gelingt, diesen Effekt in brauchbarer Form zu reproduzieren.



Sord M5

Obwohl die Standardausführung dieser Maschine nur über 4 KByte RAM verfügt, lassen sich aufgrund der hervorragenden Grafikfähigkeiten außerordentlich interessante Programme schreiben.

Bis vor einiger Zeit wurden die meisten Heimcomputer in Kalifornien entworfen, inzwischen haben jedoch auch aus England stammende Produkte einen großen Anteil am Computemarkt. Es ist jedoch nur eine Frage der Zeit, bis die Japaner auch diesen Markt dominieren werden, den sie in fast allen anderen Bereichen der Konsumelektronik bereits beherrschen. Der Sord M5 ist der erste japanische Microcomputer, der den Durchbruch schaffte.

Von der Größe her ähnelt der Sord dem Sinclair Spectrum, er ist aber schwerer und in seiner soliden Bauweise und Kompaktheit auch robuster. Auch in anderen Aspekten sind sich diese beiden Geräte ähnlich, z. B. in der Z80A-Zentraleinheit, in der Tastenbelegung mit vollständigen BASIC-Funktionen und der Programm- und Datenspeicherung auf Cassette. Doch schon die eingebaute Centronics-Schnittstelle zeigt, daß der Sord in seinem Inneren über weitaus größere Möglichkeiten verfügt. Die hauptsächlichen Unterschiede zwischen beiden Geräten liegen in der Größe des RAM-Speichers, der mit vier frei verfügba-

Druckeranschluß

An dieser Stelle befindet sich ein Centronics-Druckerinterface. Somit gibt es eine breite Palette von Druckern, die mit dem M5 verbunden werden können.

TV-Signal

Hier wird ein TV-Signal ausgegeben.

Modulator

Die internen Impulse des Computers werden in ein TV-Signal umgewandelt.

Monitoranschluß

Mit dem unveränderten Video-Signal aus dieser Verbindungsbuchse kann ein Monitor betrieben werden.

Lautsprecherausgang

Das Signal dieses Ausgangs kann über Verstärker und Lautsprecher ausgegeben werden.

Videosteuerung

Der Texas TMS 9929 Video Display-Prozessor kontrolliert die Bildschirmausgabe und kann bis zu 32 unterschiedliche Sprites steuern.

Eingang für die Spielsteuerung

Hier werden zur Steuerung der Spiele die Joypads des Sord angeschlossen.

Bildschirmspeicher

Alle Daten für die Steuerung des Bildschirms sind in diesem Speicherbereich mit 16 KByte RAM Kapazität untergebracht.



Die ROM-Cartridge

Programmiersprachen können auf dem M5 leicht gewechselt werden, da sie sich in einer ROM-Cartridge befinden. Für den Sord sind drei BASIC-Versionen erhältlich: BASIC-I (für Anfänger), BASIC-G (speziell geeignet für Grafik) und BASIC-F (für wissenschaftlich-mathematische Programmierung). Außerdem gibt es ein spezielles anwenderorientiertes Universalprogramm mit dem Namen FALC.



Cassettenausgang

Der Cassettenausgang läuft über einen DIN-Stecker. Über diesen wird auch der Motor des Cassettenrecorders gesteuert.



Spielsteuerung

Diese 'Joypads' von Sord entsprechen den konventionellen Joysticks. Für alle vier Richtungen senden sie ein spezielles Interrupt-Signal an die Zentraleinheit.

Netzanschluß

Die Stromzufuhr des Sord erfolgt über ein Netzgerät.

Spezialchip

In den M5 ist ein Spezialchip eingebaut, dessen interne Logik die komplexen Fähigkeiten des Gerätes ermöglicht.

ROM

Im ROM befinden sich die Programme zur Steuerung der grundlegenden Maschinenfunktionen. Sie sind für die Bildschirmausgabe, die Tastatur und die Cassette verantwortlich.

Zentraleinheit

Die Zentraleinheit des Sord M5 ist der bekannte Z80A. Er wird mit 3,58 MHz getaktet.

RAM

Der dem Anwender zugängliche Arbeitsspeicher befindet sich in zwei großen Chips, die von anderen Bereichen des RAM getrennt gehalten sind.

Taktgeber

Ein Großteil der sauberen Befehlsausführung des M5 beruht auf diesem hochentwickelten Taktgeber.

ren KByte (aufrüstbar auf 36 KByte) bei dem Sord weitaus kleiner ist als bei dem Spectrum. Auf der anderen Seite wurde der Sord aber seinerzeit mit guten Grafik- und Tonprozessoren ausgestattet.

Die Grafik wird über einen TI 9918, 9928 oder 9929 (abhängig von dem Verkaufsland) gesteuert und verfügt über eine Auflösung von 192 x 256 Punkten in bis zu 16 unterschiedlichen Farben. Drei der vier Hauptgrafikarten erlauben die Bewegung von bis zu 32 unabhängigen Sprites in Standard- oder Übergröße. Der Zeichensatz der Maschine besteht aus Groß- und Kleinbuchstaben, den Satzzeichen und Zahlen sowie einer großen Anzahl von Sonderzeichen. Zusätzliche Linien- und Blockgrafik und ein Zeichengenerator ermöglichen die Entwicklung eigener Zeichensätze.

**SORD M5****PREIS**

ca. 580 DM

ABMESSUNGEN

185 x 70 x 55 mm

GEWICHT

1 kg

ZENTRALEINHEIT

Z80A

TAKTFREQUENZ

3,58 MHz

SPEICHERKAPAZITÄT

8 KByte ROM. 20 KByte RAM, von denen 16 KByte für grafische Darstellung reserviert sind. Mit Cartridges kann das ROM auf 16 KByte und das RAM auf 32 KByte erweitert werden.

BILDSCHIRM-DARSTELLUNG

Bis zu 16 Farben, die auf unterschiedlichen „Ebenen“ dargestellt werden können. Es gibt Sprite-Grafiken und vier verschiedene Darstellungsarten: zwei für die Grafik, eine für Text und eine für den „Farbenmodus“.

INTERFACE

Cassette, Drucker (Centronics), Joypad-Anschluß, ROM-Cartridge, Lautsprecheranschluss

MITGELIEFERTE PROGRAMMIERSPRACHEN

Die Sprachcartridge enthält Integer-BASIC und BASIC-I.

WEITERE PROGRAMMIERSPRACHEN

BASIC-G (Grafik), BASIC-F (Fließkomma-BASIC), FALC (Kalkulations- und Datenverwaltungssystem)

ZUBEHÖR

Netzgerät, Cassettenkabel, Fernseher-Verbindungskabel, 2 Joypads mit Kabel, Cartridge mit BASIC-I und eine Cassette mit zwei Spielen

TASTATUR

55 Tasten

Andere Maschinen wie z.B. der TI99/4A verwenden ebenfalls diese Art von Grafikprozessoren, die den Sord trotz seines kleinen RAM-Bereiches zu einem wirkungsvollen Gerät werden lassen. Da der Bildschirmspeicher vollständig von dem Programmspeicher getrennt ist, befinden sich im Hauptspeicher nur das eigentliche Programm und die Daten, die dafür benötigt werden.

MSX-Standard

Im Gespräch ist seit einiger Zeit der ‚MSX‘-Standard für Heimcomputer, der von einer Gruppe führender japanischer Elektronikhersteller – Sord eingeschlossen – entwickelt wurde. Da sich voraussichtlich alle MSX-Hersteller an diesen Industriestandard für Heimcomputer halten werden (im Bereich der Hardware und Software), ist es möglich, Programme zu schreiben, die auf unterschiedlichen Fabriken ohne spezielle Anpassungen laufen.

Im Bereich der Tonerzeugung bestimmt der MSX-Standard, daß ein spezieller Prozessor verwendet werden muß. In den Sord M5 ist jedoch der TI 76489 eingebaut, da mit ihm die Tonerzeugung besser als mit dem GI-Prozes-

sor gesteuert werden kann, obwohl auch hier drei Tonkanäle und ein Geräuschkanal zur Verfügung stehen. Das bedeutet, daß der M5 dem MSX-Standard nicht ganz gerecht wird.

Drei unterschiedliche BASIC-Versionen, verschiedene Verwaltungsprogramme (Utilities) und einige Spiele werden in Form einer ROM-Cartridge geliefert, die über bis zu 16 KByte verfügen kann. Der M5 ist zwar teurer als andere Geräte dieser Leistungsgruppe, seine Fähigkeiten rechtfertigen jedoch den Preis.

Die mitgelieferte Dokumentation umfaßt den ‚User's Guide‘, in dem erklärt wird, wie der Computer mit dem Fernseher und weiteren Peripheriegeräten verbunden wird, sowie ein 100 Seiten starkes BASIC-Handbuch. Dieses enthält neben den üblichen Erklärungen der einzelnen Tasten und deren Belegung auch einige Listings, mit denen auch der Anfänger erste Spielprogramme auf den Bildschirm zaubern kann. Alle BASIC-Befehle und Funktionen werden anhand von Beispielen erläutert. Im Anhang werden dann noch die wichtigsten Ein/Ausgabefunktionen, die Bildschirmansteuerung, Zeichenbelegung sowie einige Grafikbeispiele aufgezeigt. Anhang G enthält eine Tabelle der ASCII-Codes.



Die Tastatur des Sord M5 besteht aus Gummitasten. Über insgesamt 55 dieser Tasten können mit Hilfe einer Funktionstaste alphanumerische Zeichen, grafische Symbole oder ganze BASIC-Befehle eingegeben werden. Hält man die Tasten heruntergedrückt, so wiederholt sich das entsprechende Zeichen automatisch (Auto-Repeat-Funktion).

Welche Sprachen verstehen Computer?

?

■ Wieviele Arten der Datenübertragung gibt es?

Man unterscheidet zwischen drei verschiedenen Arten der Datenübertragung – neben Simplex-Betrieb gibt es Halb- und Duplex-Betrieb. Beim sogenannten Simplex-Betrieb werden die zu übertragenden Daten beispielsweise über einen Übertragungsweg von einem Computer zu einem Peripheriegerät geschickt. Dabei dient ein Gerät als Sender, das andere als Empfänger. Der Datenfluß findet also lediglich in einer Richtung statt. Im Gegensatz dazu steht das Halb-Duplex-Verfahren, wobei die beiden Geräte abwechselnd die Rolle des Empfängers und Senders übernehmen. Anders als beim Simplex-Verfahren wird hier in beiden Richtungen übertragen. Da die Übertragung wechselweise stattfindet, wird dieser Vorgang Wechselbetrieb genannt. Beim Duplex-Verfahren wiederum findet der Datenverkehr auf zwei parallelen Datenkanälen statt. In puncto Schnelligkeit ist dieses Verfahren den beiden anderen, Simplex und Halb-Duplex, klar überlegen. Der Duplex-Betrieb wird übrigens auch

Gegen-Betrieb und Voll-Duplex-Betrieb genannt.

?

■ Was ist der Unterschied zwischen Einzelblatt- und Endlospapierzuführung?

Für den Einzelblatteinzug, auch Formulareinzug genannt, ist eine spezielle Mechanik, die auf den Drucker gesetzt wird und ihn mit einzelnen Blättern versorgt, notwendig. Doch nicht jeder Drucker verfügt über die technischen Voraussetzungen, Einzelblatteinzug betreiben zu können. Man unterscheidet ferner zwischen Einzelblatteinzügen für eine oder mehrere Sorten Papier (Einschacht- oder Zweischachtmechanik). Die Hersteller von Druckern bieten Einzelblatteinzüge meist als Zusatzgerät an.

Endlospapier dagegen sind zusammenhängende Papierbögen, die auf beiden Seiten gelocht sind. Das Papier wird über eine mit Stachelwalzen bestückte Traktorführung in den Drucker eingezogen. Die Seiten sind in Zickzackform verbunden.

?

■ Was ist eigentlich ein „Fenster“?

Mit der sogenannten Fenstertechnik werden Programme anwenderfreundlicher gemacht. Vergleichbar mit einzelnen Seiten, die man in einem Hefter (Bildschirm) ablegt, werden in verschiedenen Fenstern verschiedene Programme abgelegt, die dem Benutzer ständig zugänglich sind. Zwischen mehreren Fenstern oder Programmen ist ein Austausch der Daten möglich. Man könnte sich in einem solchen Fall ein Bilanzprogramm vorstellen, das mit Grafik versehen werden soll, um die verschiedenen Zahlengrößen grafisch, beispielsweise als Blöcke, darzustellen. Man unterscheidet zum Beispiel zwischen Text- und Grafikfenstern.

?

■ Warum muß man mit den meisten Computern englisch sprechen?

Die Weltsprache Englisch wird in vielen Ländern gesprochen und in den meisten anderen zumindest verstanden. Sie ist die einzige auf der gesamten Erde gebräuchliche Sprache. Computerhersteller konstruieren die Geräte mit englischen Benennungen für Funktionstasten und Ein- und Ausgänge. Auch Programme werden mit Befehlen gestaltet, die aus dem Englischen stammen. Auf diese Weise wird erreicht, daß nur eine Art von Geräten und Programmen erstellt werden muß, und trotzdem kann der Computer in der ganzen Welt verkauft werden – in Norwegen wie in Japan. Einige Programmiersprachen bieten jedoch die Möglichkeit, bestimmte Befehle in einer anderen Sprache, zum Beispiel in Deutsch, zu definieren.

Auf der ganzen Welt, auch bei Computerbenutzern, hat sich Englisch als Weltsprache durchgesetzt. Die meisten Programme enthalten einheitliche Befehle in englischer Sprache.



Geordnete Dateien

Computer sind in der Lage, Daten optimal zu ordnen, zu speichern und auf Abruf in korrekter Reihenfolge auszugeben. Zwei Beispiele demonstrieren, wie man diese Fähigkeiten optimal nutzen kann.

Eine Norddeutsche Firma hat Verkaufsstellen in sechs verschiedenen Städten, zu denen sie einmal im Monat einen Lastwagen mit Waren schickt. Der Computer der Firma errechnet den kürzesten Weg:

Stadt\$(1)	„Hamburg“
Stadt\$(2)	„Hannover“
Stadt\$(3)	„Neumünster“
Stadt\$(4)	„Flensburg“
Stadt\$(5)	„Bremen“
Stadt\$(6)	„Kiel“

Die Namen der Städte sind auf einem sequentiellen Band willkürlich gespeichert. Sie werden vom Band in die Matrix ‚Stadt\$()‘ geladen. Der Computer findet heraus, daß folgende Reihenfolge die beste ist:

„Flensburg“	Stadt\$(4)
„Neumünster“	Stadt\$(3)
„Kiel“	Stadt\$(6)
„Hamburg“	Stadt\$(1)
„Hannover“	Stadt\$(2)
„Bremen“	Stadt\$(5)

Die Namen der Städte werden jetzt nicht nochmals in dieser Reihenfolge gespeichert, sondern nur die Positionsnummern der Matrix ‚Stadt\$()‘. Damit wird vermieden, daß Speicherplatz verschwendet wird.

4	„Hamburg“
5	„Hannover“
2	„Neumünster“
1	„Flensburg“
6	„Bremen“
3	„Kiel“

Die Zahlen links sind der Index der Matrix ‚Stadt\$()‘. Der Vorteil dieser Indizierung liegt darin, daß die ursprünglichen Daten nicht sortiert oder kopiert werden müssen, sondern nur die Indizes.

Ein wohlgeordneter Karteikasten ist ein nützliches Hilfsmittel, um die Übersicht zu behalten. Kommen die einzelnen Karteikarten jedoch durcheinander, so wird dieses System sofort nutzlos. Die Grundlage eines Informationssystems ist folglich nicht die darin enthaltene Information, sondern ihre Organisation. Als Beispiel soll ein Eintrag in einem Branchenverzeichnis dienen:

Schmidt, H., Hauptstraße 15, Essen.

An sich ist der Informationswert dieser Adresse gering. Wenn man jedoch weiß, daß sie dem Kapitel über Bäcker entstammt, dann erhält diese Adresse eine neue Bedeutung.

Die einfachste Datenstruktur ist eine Ansammlung von Daten, die alle die gleichen Merkmale enthalten. Der Name dieser Datensammlung sagt etwas über die darin enthaltene Information aus. Datensammlungen dieser Art können die Arbeit sehr vereinfachen. Eine Datei kann als eine große Informationseinheit angesehen werden oder als eine bestimmte Gruppierung kleinerer Informationseinheiten. Ist eine Datei sehr groß, kann das Auffinden einer bestimmten Information bedeuten, daß die ganze Datei von Anfang an Stück für Stück durchsucht werden muß, bis die entsprechende Information gefunden ist. Diese Arbeitsweise wird sequentielles Suchen oder sequentieller Zugriff genannt; die entsprechende Datei ist eine sequentielle Datei. Sequentielle Dateien sind nicht nur weit verbreitet, weil sie billig und praktisch sind, sondern auch weil sie die Methodik des menschlichen Denkens reflektieren. Diese Methode hat aber auch den Nachteil, daß sie umständlich und langsam zu handhaben ist, so daß große Datensammlungen oft in Abschnitte unterteilt werden, auf die direkt zugegriffen werden kann, ohne daß die gesamte Datei durchsucht werden muß. Als Vergleich lassen sich Bücher aufführen, die einfache sequentielle Dateien darstellen. Unter Verwendung von Kapiteln, Seitenzahlen und Inhaltsverzeichnissen hat sich ihre Struktur jedoch gewandelt. Die Kapitel sind jetzt Unterabteilungen des Buches, und die einzelnen Seiten sind Unterabteilungen sowohl des Kapitels als auch des betreffenden Buches.

Eine Datei, die keine sequentielle Suche erfordert, wird Datei mit direktem Zugriff genannt. So ist zum Beispiel eine Ansammlung von Musikstücken auf einem Tonband eine se-

quentielle Datei; die gleiche Sammlung auf einer Langspielplatte ist eine Datei mit Direktzugriff: Auf dem Band kann ein Musikstück nur gefunden werden, wenn man es von Anfang an abspulen läßt, ein Abschnitt auf einer Langspielplatte dagegen kann direkt angesprochen werden, indem die Nadel direkt über den Anfang des entsprechenden Musikstückes geführt wird.

Direkt oder sequentiell

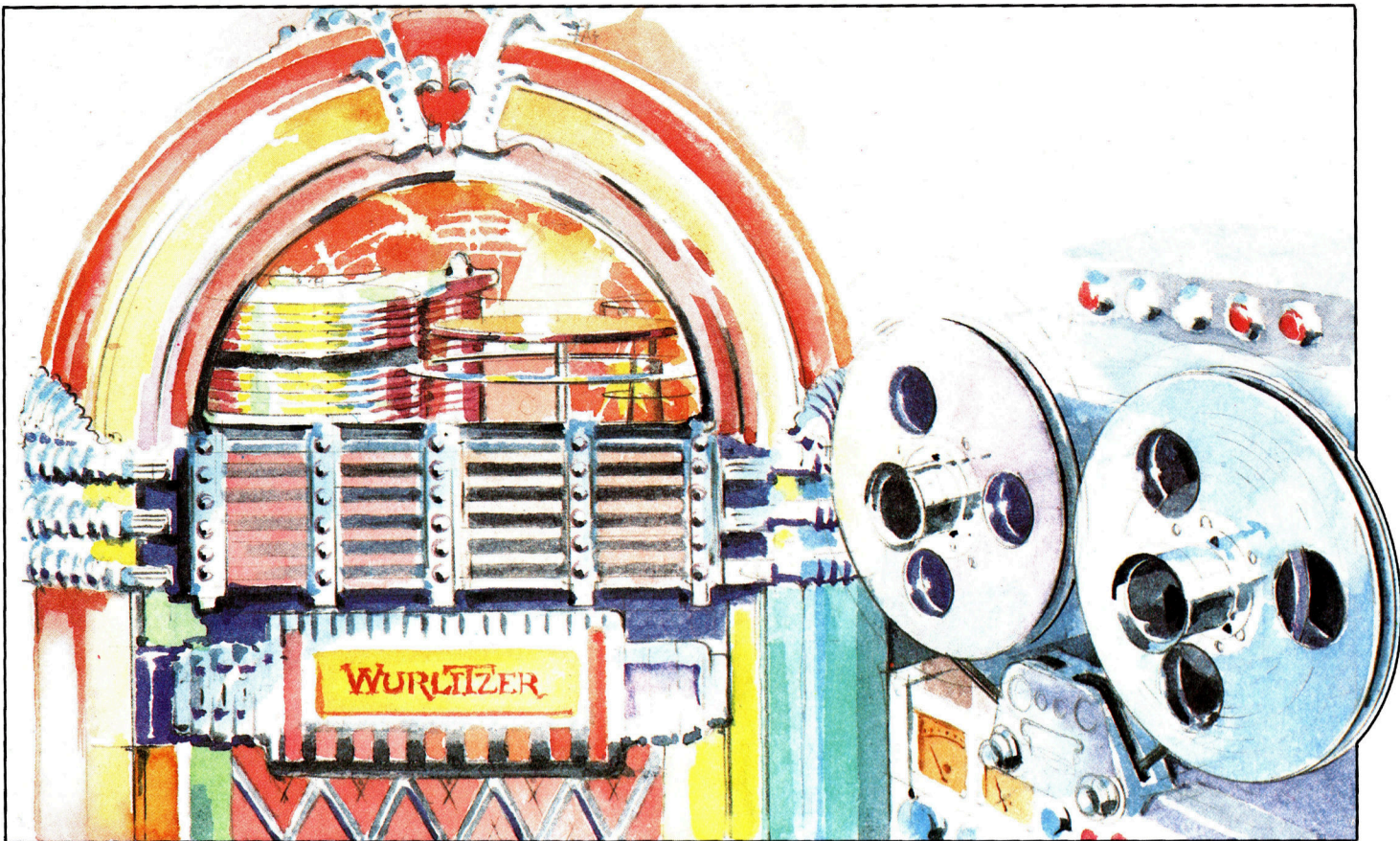
Direktzugriff hängt davon ab, daß man genau weiß, wo welche Daten abgelegt sind. Dieses Wissen bedeutet Arbeit und kostet Geld. Computer können große Mengen von Informationen schnell verarbeiten und verwenden dafür eine ganze Reihe von Datenstrukturen. Daten werden auf externen Speichermedien z. B. auf Diskette oder Band in strukturierter Weise oder sequentiell gespeichert, im Arbeitsspeicher des Computers sind sie jedoch auf völlig andere Weise abgelegt.

Nehmen wir beispielsweise an, eine Bäckerei hat die Adressen all ihrer Verkaufsstellen in einer sequentiellen Datei auf einem Cassettenband gespeichert und möchte nun die Lieferpläne für die Fahrer erstellen, die das Brot ausfahren. Auf dem Band kann die Datei folgendermaßen aussehen:

Andersen Hauptstr. 22
 Braun & Co Seitenweg 108
 Ehlers Uferstr. 49
 Wilhelm Hauptstr. 7
 Wagner Rathausstr. 65
 Zink Rolandsweg 31

Wird die Datei vom Band in den Arbeitsspeicher gelesen, dann erhält jeder Name und jede Adresse eine eigene numerierte Speicherstelle. All diese Speicheradressen sind in einem Speicherblock zusammengefaßt, der einen eigenen Namen hat. Die Datei im Hauptspeicher sieht dann so aus:

BLOCKNAME: Verkaufsstellen
 1) Andersen Hauptstr. 22
 2) Braun & Co Seitenweg 108
 3) Ehlers Uferstr. 49
 4) Wilhelm Hauptstr. 7
 5) Wagner Rathausstr. 65
 6) Zink Rolandsweg 31



Jetzt kann auf jede einzelne Information direkt zugegriffen werden, indem nur der Name des Blocks und die interne Blocknummer angegeben wird. Verkaufsstelle (4) zum Beispiel enthält „Wilhelm Hauptstr. 7“. Diese Struktur wird Array oder Matrix genannt und von Computern am häufigsten zur internen Datenverarbeitung genutzt. Sie ähnelt einem Buch mit je einer Information pro Seite. Durch diese einfache Struktur verwandelt sich eine Datei mit lesbaren Namen und Adressen zu einem Block anonymer Daten. Der Computer muß lediglich wissen, wo der Datensatz ist und was er damit machen soll.

Die Daten in der Matrix „Verkaufsstelle()“ sind alphabetisch geordnet. Diese Reihenfolge ist aber mit Sicherheit nicht die, in der die Verkaufsstellen abgefahren werden sollten. Nehmen wir an, die beste Lieferroute sei folgende:

- 1) Wilhelm Hauptstr. 7
- 2) Andersen Hauptstr. 22
- 3) Ehlers Uferstr. 49
- 4) Braun & Co Seitenweg 108
- 5) Zink Rolandsweg 31
- 6) Wagner Rathausstr. 65

Diese Lieferroute könnte jetzt in einer zweiten Matrix gespeichert werden, würde damit aber den benötigten Speicherplatz verdoppeln. Computer-Besitzer wissen, daß der RAM-Bereich begrenzt ist und daß es unpraktisch oder

sogar unmöglich ist, Daten auf diese Weise zu duplizieren; man benötigt dafür also eine andere Methode.

Wenn die Daten der neuen Matrix nur durch die Positionsnummern der Datei „Verkaufsstellen()“ ersetzt werden, dann sieht die Lieferroute folgendermaßen aus

BLOCKNAME: Lieferroute

- 1) 4
- 2) 1
- 3) 3
- 4) 2
- 5) 6
- 6) 5

und bedeutet: Fahren Sie zuerst zu dem Geschäft, dessen Adresse in Verkaufsstelle (4) enthalten ist, dann zu Verkaufsstelle (1), dann zu Verkaufsstelle (3) und so weiter. Die einzige wichtige Information in der Route ist die Folge, in der die Läden beliefert werden müssen, und deshalb ist dies auch die einzige Information, die wir in der Matrix „Lieferroute()“ unterbringen müssen.

„Lieferroute()“ ist jetzt der Index der Matrix „Verkaufsstellen()“ für den Zweck der Lieferreihenfolge. Wenn der Computer diese Reihenfolge drucken will, nimmt er die Zahlen der Matrix „Lieferroute()“ und druckt damit die Namen und Adressen der Matrix „Verkaufsstellen()“ entsprechend aus.

In diesem einfachen Beispiel wurden Infor-

Eine Musikbox verfügt über durchschnittlich 200 Titel auf 100 Schallplatten. Zur Auswahl eines Musikstückes werden drei Tasten gedrückt. Die durchschnittliche Zeit von der Auswahl bis zum Abspielen beträgt 15 Sekunden. Das Magnetband eines Tonbandträgers kann die gleichen 200 Musikstücke enthalten. Zur Auswahl eines beliebigen Stückes muß das Band zurückgespult und die START-Taste gedrückt werden. Die durchschnittliche Zeit von der Auswahl bis zum Spielen beträgt wesentlich mehr als bei der Musikbox, einer Maschine mit Direktzugriff: Sie ist schnell und hochspezialisiert. Ein Tonbandgerät dagegen ist auf sequentiellen Zugriff angewiesen: Es ist langsam und weniger spezialisiert. Ein Cassettenrecorder benutzt die sequentielle Zugriffsmethode, während ein Diskettenlaufwerk über direkten Zugriff verfügt, selbst bei sequentiellen Dateien.



mationen – die Namen und Adressen der Verkaufsstellen – unterschiedlich strukturiert, die grundlegenden Daten aber durch die neuen Strukturen nicht verändert. Eine Datenstruktur verändert nie den Inhalt der Datensätze, sondern ordnet die Daten mit Hilfe anderer Daten.

So wie die Matrix ‚Verkaufsstellen()‘ über die Matrix ‚Lieferroute()‘ mit einem neuen Index versehen wurden, so können auch Indizes nach völlig anderen Kriterien aufgebaut wer-

den. Bei den bereits behandelten Datenbanken haben wir gesehen, daß bestimmte Informationen über Markierungen (Pointer) in den einzelnen Datensätzen angewählt werden konnten. Auf genau die gleiche Weise können wir in jedem einzelnen Datensatz der Matrix ‚Verkaufsstelle()‘ eine Markierung unterbringen, die die Position dieser Adresse in der Lieferroute anzeigt. Weiterhin könnte man die Datei erweitern, um anzuzeigen, welche Bestellungen zu welchem Laden gehören – Wagner ist beispielsweise immer mit 48 Weißbröten, 12 Vollkornbröten etc. zu beliefern.

In dem Speicher des Computers befinden sich riesige Datenmengen, die Byte um Byte dort abgelegt und durch Tausende von kleinsten Schaltkreisen verwaltet und gespeichert werden. Die Struktur, nach der die Daten aufbewahrt werden, bestimmt die CPU (zentrale Verarbeitungseinheit). Anhand der unterschiedlichen Datenstrukturen erkennt der Rechner, ob ein bestimmtes Byte als Teil einer Anweisung, als Ziffernstelle oder etwa als ein anderes Zeichen zu interpretieren ist.

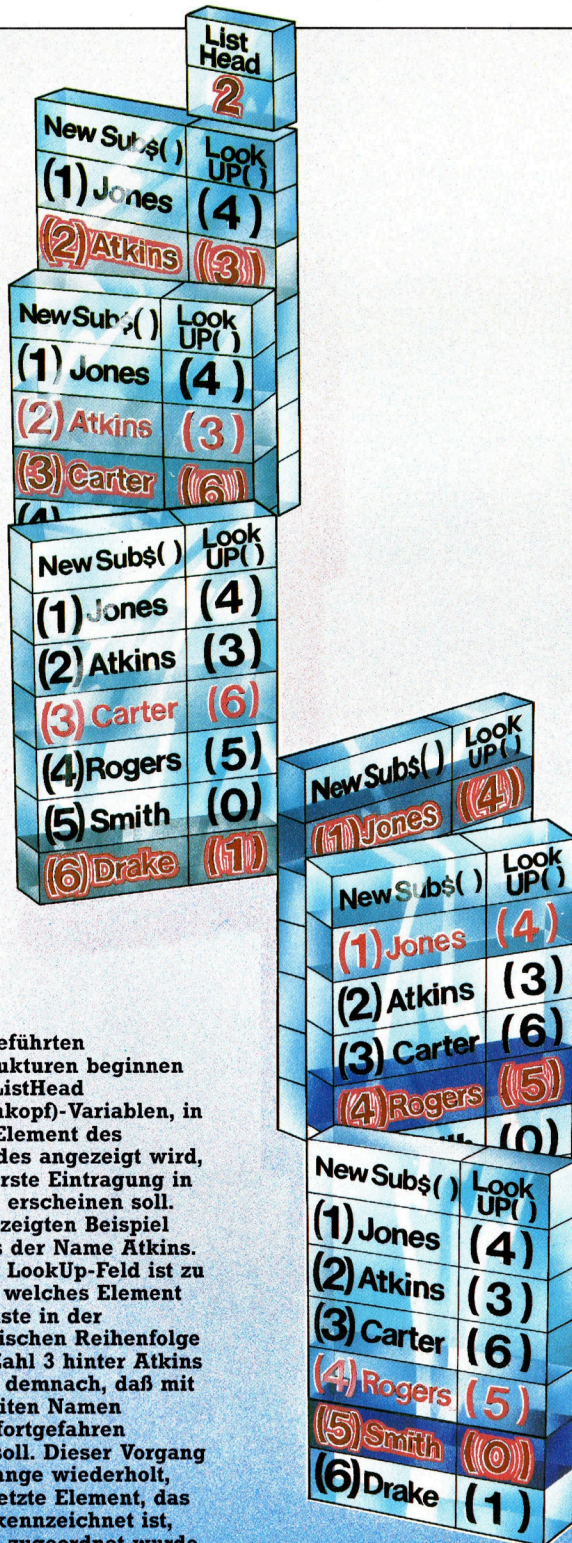
Für den Anwender ist es relativ kompliziert, diese einzelnen Datenstrukturen zu „entwirren“. Bei jeder Programmiersprache sind Art und Anzahl der Strukturen fest definiert. In BASIC werden die Daten in Form von numerischen und String-Daten abgelegt. Höhere Sprachen bedienen sich neben den genannten zusätzlich noch anderer Strukturen. Ausschlaggebend für die Qualität der Sprache ist, wie komplex und variabel diese Datenformen sind.

Variablen und Datenfelder

Wir wenden uns vorerst den beiden BASIC-Strukturen ‚Variablen‘ und ‚Datenfelder‘ zu. Indizierte Datenfelder sind vielseitig verwendbar und lassen sich einfach einsetzen. Bei Änderung der Eintragungen muß jedoch auf präzise Eingabe der Daten geachtet werden. Als Beispiel für Eingabe und Änderung der Daten soll hier ein englisches Telefonregister dienen. Die Daten müssen dabei ständig aktualisiert werden, damit auch die Eintragung der neuen Teilnehmer ins Telefonbuch sichergestellt ist. Der Übersichtlichkeit halber müssen die Namen in alphabetischer Reihenfolge aufgenommen werden, die sich jedoch verschiebt, sobald neue Namen hinzukommen. Am Montag sieht das Datenfeld ‚NewSub\$()‘ etwa so aus wie in dem untenstehenden Kasten A.

A

NewSub\$()	Index()
(1) Jones	(2)
(2) Atkins	(3)
(3) Carter	(6)
(4) Rogers	(1)
(5) Smith	(4)
(6) Drake	(5)



Die aufgeführten Datenstrukturen beginnen mit der ListHead (Tabellenkopf)-Variablen, in der das Element des Datenfeldes angezeigt wird, das als erste Eintragung in der Liste erscheinen soll. Im aufgezeigten Beispiel wäre das der Name Atkins. Aus dem LookUp-Feld ist zu ersehen, welches Element das nächste in der alphabetischen Reihenfolge ist. Die Zahl 3 hinter Atkins bedeutet demnach, daß mit dem zweiten Namen (Carter) fortgefahren werden soll. Dieser Vorgang wird solange wiederholt, bis das letzte Element, das mit 0 gekennzeichnet ist, der Liste zugeordnet wurde.

Der Index des Datenfeldes bestimmt, in welcher Folge die Namen im NewSub\$ abgelegt werden müssen, um in alphabetischer Reihenfolge zu erscheinen. In diesem Fall wäre der erste Eintrag (2) Atkins, danach würde (3) Carter folgen. Im Beispiel werden nur die Familiennamen aufgeführt: In Wirklichkeit aber sind auch die Anschrift, Vorname usw. zu berücksichtigen. Das heißt, jeder Eintrag umfaßt etwa 60 Zeichen. Diese umfangreichen Zeichenblöcke zu verschieben und zu ordnen kostet Zeit und viel Speicherplatz. Deshalb ist es sinnvoller, statt der Textblöcke nur die Indizes neu zu definieren. Nachdem der Name ‚Bull‘ in das Datenfile aufgenommen wurde, sieht die Liste so aus wie in Kasten B gezeigt.

B

ListHead(2)	
NewSub\$()	LookUp()
(1) Jones	(4)
(2) Atkins	(7)
(3) Carter	(6)
(4) Rogers	(5)
(5) Smith	(0)
(6) Drake	(1)
(7) Bull	(3)

Wie zu erkennen ist, haben sich die Eintragungen sowie ihre Reihenfolge nicht verändert. Nur im Indexfeld sind die Angaben um eine Stelle im Datenfeld nach unten gerutscht. Der neue Eintrag bewirkt im Indexfeld demnach folgendes: Die Position des neuen Namens in der alphabetischen Reihenfolge wird gesucht und gegebenenfalls werden alle nachfolgenden Elemente bis zum Listende nach hinten verschoben. Man könnte die Daten aber auch in anderer Form ordnen – und zwar indem ein neues Datenfeld – man nennt es ‚LookUp‘, das sich auf die Positionen des NewSub\$ bezieht, gebildet wird (Kasten C).

C

ListHead(2)		
NewSub\$()	LookUp()	Index()
(1) Jones	(4)	(2)
(2) Atkins	(3)	(3)
(3) Carter	(6)	(6)
(4) Rogers	(5)	(1)
(5) Smith	(0)	(4)
(6) Drake	(1)	(5)

Der erste Unterschied zu dem vorher gezeigten Beispiel besteht darin, daß eine neue Variable als ‚Tabellenkopf‘ (ListHead) definiert werden muß. Diese zeigt an, welcher Name aus NewSub\$ in der alphabetischen Reihenfolge am Anfang der Liste stehen muß. Der zweite zu beachtende Punkt ist der, daß im LookUp-Datenfeld die Zahl 0 existiert, mit der die letzte Eintragung aus NewSub\$ (Smith) markiert wird.

Aus der Abbildung ist zu erkennen, daß die numerischen Angaben vom LookUp- und Index-Feld unterschiedlich sind. Die Index-Angaben bedeuten: Das erste Element enthält NewSub\$ (2), das zweite NewSub\$ (3), das dritte NewSub\$ (6) usw. Die Angabe im ListHead dagegen liest sich so: Das erste Element befindet sich in NewSub\$ (2), wobei hier die Zahlen in LookUp die Reihenfolge bestimmen. Beendet wird die Liste mit dem Inhalt von NewSub\$ (5), der durch die Null als das letzte Element gekennzeichnet ist.

Die Index-Angabe bestimmt also die absolute Position der Elemente, während LookUp nur angibt, wo das nächste Element zu finden ist. So zeigt zum Beispiel LookUp (4) an, welcher Name nach NewSub\$ (4) als nächster in der Liste aufzunehmen ist. Die Arbeitsweise von LookUp beruht auf dem Aneinanderreihen von einzelnen Elementen, wobei Anfangs- und Endpunkt bekannt sind und nun die jeweiligen Zwischenschritte in der richtigen Reihenfolge ermittelt werden müssen.

Der Vorteil dieser Datenstruktur beruht auf ihrer Flexibilität, das heißt, es können immer wieder Daten eingefügt werden, ohne daß die Aufstellung durcheinandergebracht wird. Nachdem der Name ‚Bull‘ in der Liste aufgenommen wurde, hat sich das Datenfeld LookUp entsprechend geändert (Kasten D).

D

NewSub\$()	Index()
(1) Jones	(2)
(2) Atkins	(7)
(3) Carter	(3)
(4) Rogers	(6)
(5) Smith	(1)
(6) Drake	(4)
(7) Bull	(5)

LookUp (2), das zuvor auf NewSub\$ (3) als nachfolgendes Element der alphabetischen Liste zeigte, gibt nun NewSub\$ (7) – also den Namen ‚Bull‘ – als nächsten Listeneintrag an. Die Angabe LookUp (7), deutet auf den dem Namen ‚Bull‘ folgenden Eintrag – NewSub\$ (3) Carter – hin.

Die Abbildungen zeigen, nach welchem Schema diese Datenstruktur arbeitet. Beim vorangehenden Element muß also immer die Position des nachfolgenden angegeben werden, um die korrekte Reihenfolge zu wahren. Die Länge der Liste ist hier ohne Bedeutung. Wichtig ist dabei nur, daß beim Einfügen eines neuen Gliedes bzw. Elementes in die Kette der Kreis wieder geschlossen wird und die einzelnen Elemente mit den jeweils vorhergehenden und nachfolgenden verbunden werden. Listen in Verbindung mit Datenstrukturen sind besonders hilfreich, wenn größere Datenmengen zu verwalten und zu ordnen sind, deren Anzahl und Wertigkeiten sich häufig ändern.

Das Grafiktablett

Zum Digitalisieren von Zeichnungen und grafischen Darstellungen in eine für den Computer verständliche Form verwendet man Grafiktablets.

Was heutige Heimcomputer besonders auszeichnet, sind ihre erstaunlichen grafischen Fähigkeiten. Mit nur wenigen Befehlen lassen sich zum Beispiel einfache Zeichnungen und geometrische Formen konstruieren oder Farbänderungen vornehmen.

Lupe
Mit der Lupe werden alle zu digitalisierenden Linienzüge abgefahren.

Noch gibt es keine Möglichkeit, Zeichnungen direkt vom Papier in den Computerspeicher zu laden. Mit den bekannten Grafikvorrichtungen, wie etwa dem Lichtgriffel, lassen sich die auf den Computer übertragenen Darstellungen verändern und optimieren.

Am häufigsten findet die Computergrafik Verwendung bei der Konstruktion von Autos, Flugzeugen und Microprozessoren, aber auch bei Innenarchitekten und Modedesignern. Ist der Rohentwurf der Zeichnung sicher gespeichert, können Veränderungen vorgenommen werden, ohne daß die abgelegten Daten verloren gehen. Zuerst mußte für diesen Vorgang ein Eingabegerät entwickelt werden, das die gezeichneten Linien und Kurven in eine für den Computer verständliche Form umsetzt.

Das Grafiktablett ist solch ein Werkzeug. In der Industrie kennt man es schon fast so lange wie den Computer selbst. Für Heimcomputer gibt es erschwingliche, jedoch nicht mit den professionellen Tablett vergleichbare Modelle erst seit etwa zwei Jahren. Hochpräzise Grafiktablets, auch „Digitiser“ genannt, digitalisieren analoge Linienzüge oder grafische Darstellungen: Sie setzen diese in digitale Informationen um. Die besten Geräte haben eine Auflösung von ungefähr 0,25 Millimeter, was für Konstrukteure und Zeichner eine ausreichende Genauigkeit zuläßt.

Diese Digitiser bestehen aus einem Tablett, auf das die vorgefertigte Zeichnung gelegt wird und einem Stift (Griffel) als Eingabeein-

Fadenkreuz
Linse und Fadenkreuz bestimmen die Position des zu digitalisierenden Punktes bis auf 0,25 mm genau.

Dateneingabe
Mit einem Druckschalter signalisiert der Tablettbenutzer dem Computer, wann er die Daten eines digitalisierten Punktes speichern soll. Die meisten Lupen haben mehrere Druckschalter.

Sendespule
Die Spule sendet ein Hochfrequenzsignal, das die Gitterdrähte in der unmittelbaren Umgebung empfangen.

heit. Auf dem Tablett wird entweder direkt gezeichnet, oder die Linien der Zeichnung werden mit dem Griffel nachgezogen. Der Griffel kann ein normaler Kugelschreiber sein oder ein elektronischer Stift. Die Position des Griffels wird gelesen, und die Koordinaten werden in digitale Signale umgewandelt, die der Computer verarbeiten kann.

Zwei der präzisesten Systeme, das eine beruht auf Magnet-, das andere auf Kondensatoren-Basis, verwenden ein Drahtgitter, das in das Tablett integriert ist. Anstelle des Griffels tritt beim magnetischen System eine bewegliche Lupe mit einem Fadenkreuz. Die Spule, die das Fadenkreuz umgibt, sendet ein schwaches Hochfrequenzsignal aus, das vom Drahtgitter empfangen wird und die Tablett-Elektronik in entsprechende digitale Positionssignale

Schnittstelle
Grafiktablett und Computer sind gewöhnlich über eine serielle oder parallele Schnittstelle verbunden.

Arbeitsfläche

Auf der Arbeitsfläche wird die Zeichnung befestigt, damit sie sich nicht verschiebt, wenn man ihre Linien mit der Lupe oder einem Griffel abfährt.

umwandelt. Beim zweiten System dagegen werden codierte Signale in das Gitter eingespeist und durch Bewegen des Griffels über das Tablett von diesem umgewandelt.

Das akustische System verwendet anstelle des Gitters zwei Mikrofone. Berührt der Griffel das Tablett, so erzeugen winzige statische Entladungen nicht wahrnehmbare akustische Wellen, die die Mikrofone empfangen. Die Elektronik mißt die Zeit, die verstreicht, bis die



Am meisten verbreitet sind Grafiktablets in der Kartographie zum Sammeln von geographischen Daten. In digitaler Form gespeichert, dienen sie z. B. zur Computeranalyse vermuteter Ölvorkommen.

Tablett-Elektronik

Auf der Leiterplatte befinden sich ein Mikroprozessor und verschiedene ROM- und RAM-Bausteine. Sie wandeln die X- und Y-Koordinaten in digitale Daten für den Computer.

Empfangsantenne

Eingebettet im Bodenteil befindet sich ein Gitter isolierter Drähte. Sie sind Antennen für das HF-Signal, das von der Sendespule der Lupe kommt. Die Auflösung ist viel größer, als der Abstand der Drähte vermuten läßt. Dies liegt an der Fähigkeit, aus der Signalgröße gegenüberliegender Drähte die Fadenkreuz-Position bestimmen zu können.

Welle beide Mikrofone erreicht. Die gemessenen Werte werden in die Koordinaten der Stiftposition umgesetzt.

Entgegen den anderen Techniken gehört das druckempfindliche Grafiktablett nicht zu den Spitzengeräten. Die Auflage besteht aus zwei elektrisch leitenden Folien, die durch eine gitterförmige Zwischenlage getrennt sind. In die Folien werden zwei unterschiedliche Hochfrequenzsignale eingespeist. Wird mit dem Griffel eine Linie gezogen, so empfängt er ein Signal, sobald sich die Folien unter dem aufgewendeten Druck berühren. Dieses Signal dient zur Bestimmung der Griffelposition. Der Nachteil dabei ist, daß die Folien durch zu starken Druck beschädigt werden.

Der einfachste und preisgünstigste Digitiser ist der sogenannte Pantograph. Das Prinzip dieses Zeichengerätes beruht auf einer vor längerer Zeit entwickelten Technik, die zum Vergrößern und Verkleinern von Zeichnungen dient und auch unter dem Namen „Storchenschnabel“ bekannt ist. Das Schulter- und Ellbogengelenk des Abtastarms sind mit Potentiometern versehen, die eine dem Verstellwinkel proportionale Spannung abgeben. Diese Spannungen werden in digitale Werte für die X- und Y-Koordinaten der Position umgewandelt.

Grafiktablets, die auf dem optischen Meßverfahren beruhen, nutzen ein aus Infrarotstrahlen gebildetes Gitter für die Ortung der jeweiligen Position. Aufgrund ihrer geringeren Sensibilität lassen sich mit diesen Geräten keine hochauflösenden Grafiken erstellen. Diese Technik, mit der einige interaktive Bildschirme arbeiten, kann beispielsweise dazu benutzt werden, Befehle aus dem Programm-Menü mit einem Fingerdruck auszuwählen.

Die vom Grafiktablett oder Digitiser gelieferten Daten muß der Computer in Bildschirmdarstellungen umsetzen. Diese Umsetzung erfolgt durch die entsprechende Software. Die Fähigkeiten eines Grafiktablets beschränken sich jedoch nicht auf das Digitalisieren der eingegebenen Daten. Alle Linien, Formen und Farben können verändert werden.

Wir stellen vor: Sound . . .

Sound und Grafik ist die erste Folge einer neuen Serie, in der untersucht wird, wie Tongeneratoren und Grafiksysteme der Computer funktionieren und was sie leisten.

Die Qualität und Anzahl von Spielprogrammen, die für einen bestimmten Heimcomputer verfügbar sind, haben inzwischen entscheidenden Einfluß darauf, ob ein Computer beim Publikum ankommt oder nicht. Neben der Verbesserung und Verfeinerung der Farbgrafik wurde auch die Sounderzeugung entscheidend weiterentwickelt. Programmierer erfolgreicher Spiele räumen der Klanguntermalung einen hohen Stellenwert ein, denn der geschickte Einsatz von Sound- und Musikeffekten erhöht entscheidend den Unterhaltungswert von Arcadespielen.

Befehle zur Tonsteuerung

Aber auch unabhängig von Spielanwendungen kann man das eigene Musikverständnis mit Hilfe des Tongenerators eines Heimcomputers vertiefen. In vielen Fällen enthält das mitgelieferte BASIC eine Reihe von Befehlen zur Tonsteuerung, die – in kurze Programme eingebaut – komplexe Tonfolgen erzeugen oder Akkorde spielen können. Einige Computer verfügen auch über Möglichkeiten, die Stimmung der Töne abzuwandeln, um sie dem Klang echter Musikinstrumente anzupassen und so für das Ohr angenehmer zu gestalten. Mit entsprechender Software können die Tasten der Tastatur umbelegt (konfiguriert) werden, um dann wie bei einem Klavier Töne im „Echtzeitmodus“ zu erzeugen.

Selbst mit nur wenig Programmiererfahrung können Sie kurze und einfache Programme schreiben, die komplizierte Klanggebilde entstehen lassen. Und sogar ohne jegliche Erfahrung lassen sich mit den Programmen der entsprechenden Softwarehäuser Melodien schreiben und spielen, die die Klangeigenschaften Ihres Computers voll ausnutzen.

Oszillatoren

Oszillatoren sind elektronische Schaltungen, die sich wiederholende Signalfolgen erzeugen. Werden diese Signale verstärkt und an einen Lautsprecher weitergegeben, erzeugen

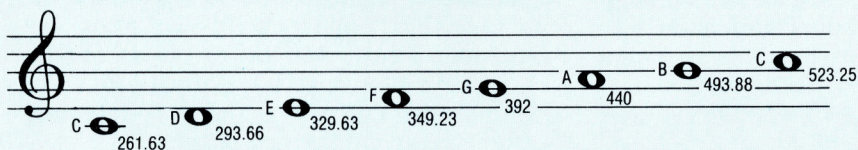
sie Töne in der entsprechend vorgegebenen Höhe. In Heimcomputern sind zwischen einem und vier Oszillatoren eingebaut. Je mehr Oszillatoren vorhanden sind, desto mehr unterschiedliche Töne können zur gleichen Zeit gespielt werden.

Frequenz

Die Frequenz bestimmt die Tonhöhe und ist damit die wichtigste Komponente. Mit Frequenz wird die Anzahl der Wiederholungen eines Signals pro Sekunde bezeichnet. Sie wird in Hertz (Hz, Zyklen pro Sekunde) gemessen. Das menschliche Ohr kann Klänge zwischen 20 und 20 000 Hz wahrnehmen. Aber auch wenn uns Töne unter 20 Hz nicht mehr zugänglich sind, können diese tiefen Frequenzen dazu verwendet werden, den Charakter der hörbaren Klänge zu verändern. Diese Technik wird Modulation genannt.

Bei der Klangerzeugung gibt es von Computer zu Computer große Unterschiede in der Bedienung. Einige haben BASIC-Befehle, die die Frequenzen unmittelbar umsetzen, so daß nur ein bestimmter Tonhöhen Schlüssel oder auch nur das Symbol der Note (A, A#, B, etc.) angegeben werden muß. Andere BASIC-Versionen sind weitaus komplizierter zu bedienen, da die entsprechenden Frequenzen erst in einer Tabelle des Bedienerhandbuchs gesucht und der Frequenzwert über POKE in die entsprechende Speicherstelle geladen werden muß. Die Tabellen zeigen meist die Frequenzen der C-DUR-Tonleiter ausgehend von dem mittleren C an. Für Programmierer, die Musik über den Maschinencode steuern wollen, sind diese Tabellen sehr nützlich, da bei dieser Art der Programmierung kein BASIC-Befehl die Umrechnung der Frequenzen erleichtern kann.

Die Frequenz jedes Tones einer Tonleiter läßt sich errechnen, indem man die Frequenz des darunterliegenden Halbtönen mit der Zahl 1,0594631 multipliziert. Das mag sich seltsam anhören, führt man aber diese Multiplikation insgesamt zwölfmal aus, so erhält man das Doppelte der ursprünglichen Frequenz. In einer Oktave gibt es 12 Halbtöne, und die Verdopplung der Frequenz läßt einen Ton eine Oktave höher erklingen. In der Skala unten sind die Frequenzen der Noten vom mittleren C aus aufwärts aufgeführt.



Bitmuster								Pixelmuster							
128	64	32	16	8	4	2	1								
1	0	0	1	1	0	0	1								
0	1	0	1	1	0	1	0								
0	0	1	0	0	1	0	0								
0	0	0	1	1	0	0	0								
0	0	1	0	0	1	0	0								
0	0	1	0	0	1	0	0								
0	1	1	0	0	1	0	0								
0	1	1	0	0	1	1	0								



... und Grafik

Sieht man sich auf dem Bildschirm die Zeichen genauer an, so bemerkt man, daß diese aus einer Anzahl kleiner Quadrate zusammengesetzt sind. Diese Quadrate werden Bildelemente (Englisch: picture elements) oder kurz ‚Pixel‘ genannt. Die meisten Heimcomputer formen ihre Zeichen innerhalb einer quadratischen Fläche, die aus 64 Pixeln besteht, denen acht Reihen mit je acht Pixeln zugeordnet sind. So kann der Buchstabe „A“ zum Beispiel aus einem Pixelmuster zusammengesetzt werden, wie wir es im untenstehenden Bild sehen. Im Speicher des Computers wird jedes leuchtende Pixel als 1 gespeichert und jedes dunkle Pixel als 0. Acht Bits ergeben ein Byte und somit nimmt jede Pixelzeile des Zeichens genau eine Speicherstelle innerhalb des Computers ein. Es werden also acht Speicherstellen zur Unterbringung eines einzigen Zeichens benötigt.

Grafische Darstellungen können aus Blöcken aufgebaut sein, die aus der ganzen Fläche, der Hälfte oder einem Viertel eines Standardzeichens bestehen. Diese Art der Darstellung wird ‚Blockgrafik‘ oder ‚Grafik mit niedriger Auflösung‘ genannt. Fast alle Heimcomputer verfügen heute über die Möglichkeit, grafische Darstellungen aus einzelnen Pixeln aufzubauen. Diese Methode ist als hochauflösende Grafik bekannt. Den Unterschied zwischen diesen beiden Methoden macht die untenstehende Sinuskurve deutlich, die beide Grafikarten darstellt.

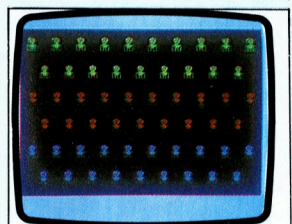
Zeichengenerierung

Für interessante und ungewöhnliche Bildschirmdarstellungen werden oft Zeichen benötigt, die von dem normalen alphanumerischen Zeichensatz abweichen. Einige Computer verfügen über einen speziellen Zeichensatz, der von der Tastatur aus direkt angesprochen werden kann. Aber auch diese Sonderzeichen genügen nicht allen Anforderungen. Auf vielen Heimcomputern kann man daher einen eigen-

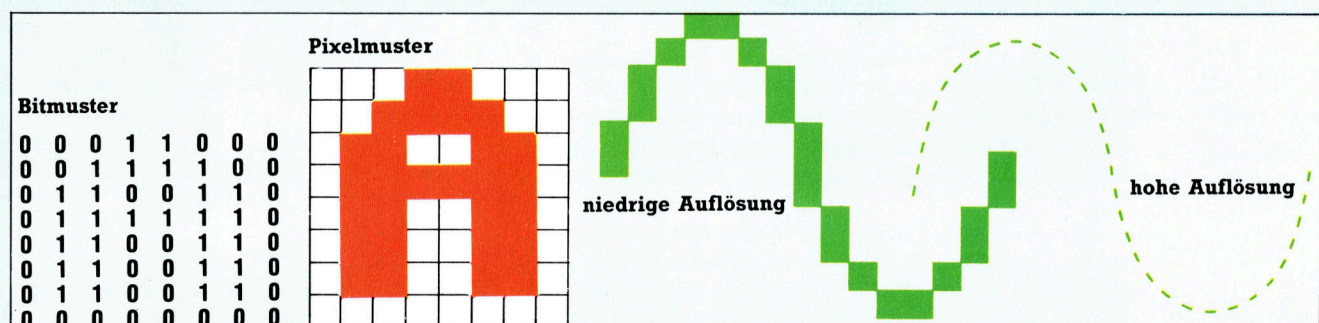
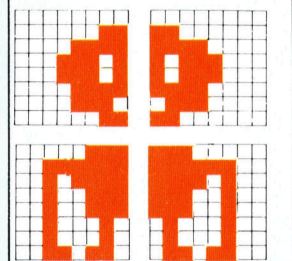
nen Zeichensatz erzeugen. Hierbei wird zu- meist das Bitmuster der acht Speicherstellen umgestellt, die ein Standardzeichen enthalten. Das neue Zeichen kann jetzt per Tastendruck und in PRINT-Befehlen genauso auf den Schirm gebracht werden wie das alte Zeichen. Das Bild auf Seite 240 unten rechts ist ein Beispiel für ein neudefiniertes Zeichen; daneben das entsprechende Bitmuster. Die Leichtigkeit, mit der neue Zeichensätze definiert werden können, variiert von Computer zu Computer. So braucht man z. B. bei dem Sinclair Spectrum das neue Bitmuster nur über den USR-Befehl einzugeben, während beim Commodore 64 zunächst der gesamte Zeichensatz von dem ROM-Speicher in den RAM-Bereich geladen werden muß, bevor über POKE die Dezimalwerte der acht Bitmuster auf die entsprechenden Speicherstellen gelegt werden können. Inzwischen aber haben unabhängige Softwarehäuser Programme zur Zeichengenerierung auf den Markt gebracht, die die Definition eines eigenen Zeichensatzes wesentlich vereinfachen.

Will man Figuren erzeugen, die über die Größe eines Standardzeichens hinausgehen, so hat man die Möglichkeit, mehrere neudefinierte Zeichen zu einer größeren Figur zusammenzusetzen. Die „Marsmännchen“ (rechts im Bild) bauen sich aus vier selbstdefinierten Zeichen auf. Das auf den Commodore 64 zugeschnittene Programm bringt diese Zeichen- gruppen in drei unterschiedlichen Farben auf den Bildschirm. In dem Programm wurde zunächst der Standardzeichensatz vom ROM in das RAM transferiert und die benötigten Grafikzeichen durch das Einlesen von Dezimalzahlen aus DATA-Zeilen per POKE in den entsprechenden Speicherstellen ausgetauscht.

Selbst wenn ein Computer das Arbeiten mit ‚Sprites‘ unterstützt, so können dabei oft nur wenige Sprites gleichzeitig auf dem Schirm gezeigt werden. Auch in diesem Fall ist daher die Möglichkeit, neue Formen selbst generieren zu können, sehr praktisch.



Diese kleinen „Marsmännchen“ sind aus vier Zeichen aufgebaut, die einzeln generiert wurden.



Dem puren Zufall ausgeliefert

Die RND-Funktion erzeugt Zufallszahlen für Glücksspiele und statistische Programme.

RND, die Abkürzung für das Wort Random, ist eine BASIC-Funktion, die zur Ermittlung von Zufallszahlen eingesetzt wird. Da diese Funktion in den verschiedenen BASIC-Dialekten unterschiedlich geschrieben wird, kann unsere Beschreibung hier sehr wohl Abweichungen gegenüber der BASIC-Version Ihres Heimcomputers aufweisen. Die Gründe für diese „dialektischen“ Unterschiede liegen in der historischen Entwicklung der Programmiersprache BASIC.

Die meisten Programme stützen sich auf Microsoft-BASIC, kurz MBASIC. Microsoft ist eine amerikanische Firma, deren BASIC eine der ersten Versionen war, die in größerem Umfang populär wurde. Für BASIC gibt es keinen offiziellen Standard, jedoch wurde die von Microsoft entwickelte BASIC-Version als „Quasi“-Standard anerkannt.

MBASIC und die meisten neueren Versionen unterscheiden sich im wesentlichen darin, daß Heimcomputer heute leistungsfähige Grafikeigenschaften aufweisen, die damals, als MBASIC entwickelt wurde, nicht verfügbar waren.

Bei den bekannten Heimcomputern sind es unter anderem das Sinclair-BASIC (im ZX81 und Spectrum) und das BBC-BASIC, die sich größtenteils vom MBASIC unterscheiden. Auch das im TI99/4A von Texas Instruments verwendete BASIC weist einige auffällige Unterschiede auf. Wie Sie Ihr Programm erforderlichenfalls modifizieren können, ist aus unseren ‚BASIC-Dialekten‘ ersichtlich.

Den Einsatz der RND-Funktion machen wir



an folgendem Würfelspiel deutlich. Wie in früheren Programmen verwenden wir auch hier Unterprogramme, deren Vorteile im übersichtlichen Aufbau eines Programmes, der leichten Lesbarkeit und der einfacheren Fehlersuche liegen.

Das Hauptprogramm beginnt mit der Anweisung RANDOMIZE in Zeile 20. Diese Anweisung, die einige BASIC-Versionen brauchen, startet die RND-Funktion und stellt sicher, daß

sich die Sequenz der Zufallszahlen nicht wiederholt. Die in Zeile 50 aufgerufene Subroutine verwendet RND dazu, der Variablen D eine Zufallszahl zuzuordnen. Dies geschieht folgendermaßen:

```
320 LET D=INT(10*RND)
```

Diese Programmzeile muß häufig modifiziert werden (siehe BASIC-Dialekte). Zu beachten ist hier der in Klammern stehende Ausdruck, mit dem die Anzahl der Zahlen festgelegt wird. Ohne ihn (z. B. mit der Anweisung LET A=RND) würde der Wert A eine Zahl zwischen 0 und 1 sein. Da wir jedoch keine Zahl wollen, die kleiner als 1 ist, multiplizieren wir die Zahl mit 10, was durch die Anweisung LET A=10*RND geschieht. Wenn uns RND z. B. die Zahl 0,125455 präsentieren würde, wäre der Wert von A jetzt 1,25455.

INTEGER-Funktion

Da wir eine ganze Zahl ohne Dezimalstellen haben wollen, verwenden wir die Funktion INT (Abkürzung von INTEGER für „Ganzzahl“) folgendermaßen: LET A=INT(10*RND). Durch die Multiplikation und Ganzzahlbildung bietet sich auch die Möglichkeit, die Sequenz der erzeugten Zufallszahlen leicht zu ändern.

Bei einigen BASIC-Versionen kann die obere Begrenzung der zu generierenden Zufallszahlen in dem hinter RND in Klammern stehenden Ausdruck spezifiziert werden. Bei einigen BASIC-Dialekten wird auf die Anweisung PRINT RND(6) eine ganze Zahl im Bereich von 1 und 6 produziert. Man überprüft also, ob die Zufallszahl größer als 6 oder kleiner als 1 ist, weil für ein Würfelspiel nur die Zahlen in diesem Bereich notwendig sind. Diese Auswahl treffen die Programmzeilen 330 und 340:

```
330 IF D>6 THEN GOTO 320
340 IF D<1 THEN GOTO 320
```

Liegt D außerhalb des Bereiches 1 bis 6, veranlaßt GOTO das Programm, zurückzuspringen und einen neuen Versuch zu machen.

Haben wir nun eine Zufallszahl für D zwischen 1 und 6, geht die Subroutine „Würfeln“ zum Hauptprogramm zurück, das uns die Mitteilung ausgibt: IHR ERGEBNIS IST A. Dahinter folgt das Bild eines Würfels. Beachten Sie bitte, wie die entsprechende Darstellung der Würfelseite gewählt wird. Dies geschieht in der SELECT-Subroutine folgendermaßen: Ist die gewürfelte Zahl (und damit D) z. B. eine 1, ruft Zeile 410 die in Zeile 530 beginnende Subroutine in folgender Weise auf:

```
410 IF D=1 THEN GOSUB 530
```

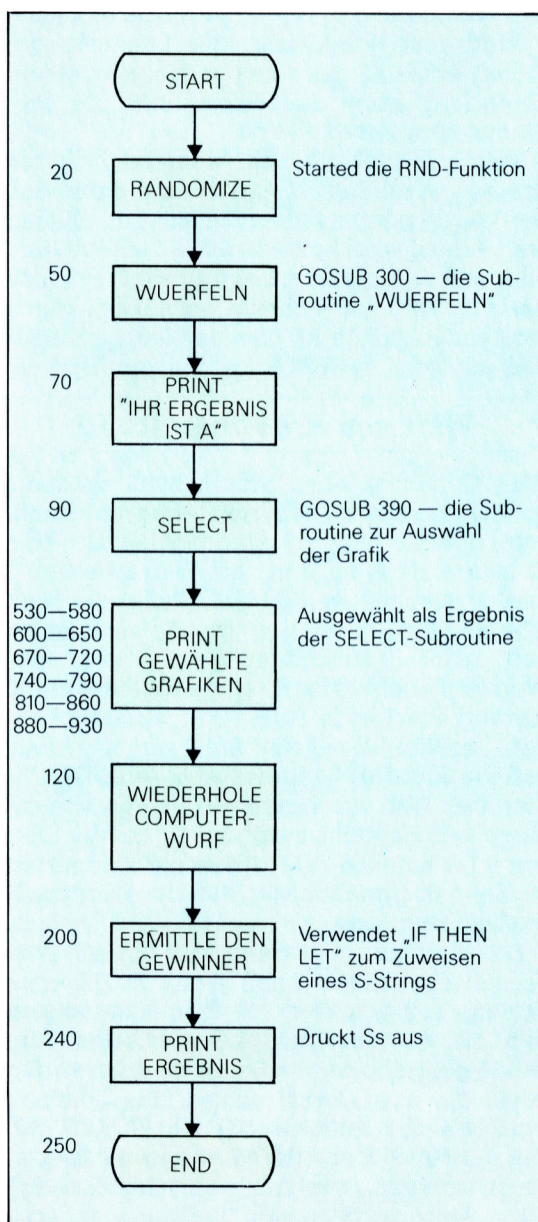
Diese Subroutine ist nichts weiter als eine Reihe von PRINT-Anweisungen, mit deren Hilfe grobskizzierte grafische Darstellungen auf dem Bildschirm erscheinen. Nachdem das Programm eine Zufallswürfelzahl gewählt hat, wiederholt es den Vorgang, um eine Würfelzahl für den Computer auszusuchen. Derjenige Teil des Programms, der entscheidet, wer gewonnen hat, ist im Hauptprogramm enthalten. Zeile 200 vergleicht die Werte von M (Mein Würfel) mit C (Würfel des Computers), um festzustellen, ob sie gleich sind. Ist dies der Fall, wird das Wort UNENTSCHIEDEN der Stringvariablen S\$ zugeordnet. Zeile 210 stellt fest, ob M größer als C ist. Wenn ja, wird der Stringvariablen S\$ das Wort GEWONNEN zugeordnet. Wird in Zeile 220 errechnet, daß M kleiner als C ist, veranlaßt das Programm, daß der Stringvariablen S\$ die Worte DER COMPUTER HAT GEWONNEN zugeordnet werden. Zeile 240 dient lediglich dazu, das Ergebnis auszugeben. Damit ist das Spiel zu Ende.

In Anbetracht dessen, daß die RND-Funktion bei vielen Computertypen geändert werden muß und einige BASIC-Versionen (wie z. B. Microsoft) die RANDOMIZE-Anweisung zum Starten einer neuen Sequenz von Zufallszahlen erfordern, stellt sich die Frage, ob es einen Weg gibt, echte Zufallszahlen (also nicht voraussagbare) zu generieren, ohne diese RND-Funktion zu benutzen. Es gibt dafür einige Techniken.

Eine der Funktionen, die wir bisher noch nicht besprochen haben, ist INKEY\$. Jedesmal, wenn das Wort INKEY\$ auftaucht, sucht das Programm die Tastatur daraufhin ab, ob eine Taste gedrückt worden ist. Dabei wartet das Programm nicht darauf, daß eine Eingabe erfolgt, wie beim Befehl INPUT. INKEY\$ wird deshalb gewöhnlich in einer Schleife angeordnet. Das Programm sucht dann so lange die Tastatur ab, bis irgend etwas eingegeben wird. In der Schleife befindet sich gewöhnlich eine Testfunktion, die die Schleife dann beendet, wenn ein geeignetes Zeichen eingegeben worden ist. Auf diese Weise kann ein Programm so geschrieben werden, daß eine Zählschleife entsteht, die dann beendet ist, wenn ein bestimmtes Zeichen eingetippt ist. Versuchen Sie einmal dieses Programm:

```
10 PRINT "DRUECKEN SIE DIE LEERTASTE"
20 FOR X=0 TO 1
30 LET R=R+1
40 LET A$=INKEY$
50 IF A$="" THEN GOTO 80
60 LET X=0
70 NEXT X
80 FOR Q=0 TO 1
90 IF R<10 THEN GOTO 130
100 LET Q=0
110 LET R=R/10
120 NEXT Q
130 PRINT INT(R)
140 END
```

Wird R eine Zufallszahl sein? Sie sollte es sein, und nun wollen wir sehen, warum. Zeile 10 druckt die Aufforderung DRUECKEN SIE DIE LEERTASTE. Noch ehe wir Zeit haben, dieser Aufforderung zu folgen, hat das Programm die Schleife FOR X=0 TO 1 in Zeile 20 aufgerufen.



Der Ablaufplan zeigt die vom Programm ausgeführten Haupttätigkeiten in vereinfachter Form. Auf der linken Seite stehen die zugehörigen Zeilennummern, auf der rechten Seite finden sich kurze Erläuterungen. Dieses Beispiel ist kein vollständiger Ablaufplan, da viele „Entscheidungen“ und Programm-Verzweigungen nicht dargestellt sind.

Die Zahlen 0 und 1 mögen der Schleife als unbekannte Grenzwerte erscheinen, aber darüber gleich mehr. Zeile 30 ordnet beim ersten Schleifendurchgang der Variablen R den Wert 1 zu. Zeile 40 weist der Stringvariablen das Zeichen zu (LET A\$), das eingetastet ist, und zwar mit Hilfe der INKEY\$-Funktion. Haben Sie beispielsweise den Buchstaben R eingetippt, wird R A\$ zugewiesen. Zeile 50 überprüft A\$ darauf hin, ob es sich um eine Leerstelle handelt, die in BASIC durch eine Leerstelle zwischen den Anführungszeichen dargestellt wird (" "). Ist A\$ eine Leerstelle, springt das Programm zu der in der GOTO-Anweisung angegebenen Zeilennummer, im anderen Fall fährt das Programm mit der nächsten Zeile fort.

Diese nächste Zeile 60 besagt: LET X=0. Damit ist X der Index der Schleife. Die NEXT-Anweisung in Zeile 70 läßt das Programm auf den Anfang der Schleife in Zeile 20 zurückkehren. Weil X auf 0 zurückgestellt wurde, wiederholt sich die Schleife FOR X=0 TO 1 so lange, wie die Überprüfung IF A\$=" " ergebnislos bleibt.

Wird nun irgendwann die Leertaste gedrückt, erhält A\$ das eine Leerstelle repräsentierende Zeichen zugeordnet und das Programm springt zu Zeile 80.

Was geschieht jedoch, während sich die Schleife wiederholt? Zeile 30 sorgt dafür, daß der Wert von R bei jedem Schleifendurchgang um 1 erhöht wird; beim ersten Schleifendurchgang würde R gleich 1, beim zweiten 1+1 usw. gesetzt. Wird die Schleife schließlich durch die Testfunktion in A\$ beendet, läßt sich ablesen, auf welchen Wert R angestiegen ist.

Wert von R kleiner als 10

Weil Computer aber sehr schnell arbeiten, kann R bereits in den Hunderten angelangt sein, ehe die Leertaste gedrückt wurde. Was ist zu tun, wenn für R nur ein Wert zwischen 1 und 10 ermittelt werden soll? Zeile 80 richtet eine andere Schleife ein, mit der wir R testen und durch 10 dividieren können, wenn der Wert größer als 10 ist. So lange R größer als 10 ist, wird der Test in Zeile 90 ergebnislos bleiben, der Wert von Q wird auf 0 zurückgestellt, und die Schleife wiederholt sich. Zeile 110 dividiert den Wert von R durch 10, und es wird solange kein Ergebnis ausgedruckt, bis der Wert von R bis auf eine Zahl kleiner als 10 reduziert ist. Zeile 30 gewährleistet, daß der Wert von R nie Null sein kann.

Das Problem liegt darin, daß BASIC zwar schnell, aber nicht schnell genug ist. Die erste Schleife läßt den Wert von R so lange ansteigen, bis er in die Hunderte oder selbst Tausende geht, ehe wir die Leertaste drücken. Solange Sie nicht darauf achten, den Zeitraum zwischen der Aufforderung DRUECKEN SIE DIE LEERTASTE und dem Drücken der Leertaste zu variieren, ist es sehr wahrscheinlich, daß die jeweils verstrichenen Zeiträume in etwa

```

10 REM WUERFELSPIEL — HAUPTPROGRAMM
20 RANDOMIZE
30 REM IHR WURF
40 REM GOSUB WUERFELN-ROUTINE
50 GOSUB 300
60 LET M=D
70 PRINT "IHR ERGEBNIS IST A"
80 REM GOSUB SELECT-ROUTINE
90 GOSUB 390
100 PRINT
110 REM COMPUTERWURF
120 REM GOSUB WUERFELN-ROUTINE
130 GOSUB 300
140 LET C=D
150 PRINT "DAS COMPUTERERGEBNIS IST A"
160 REM GOSUB SELECT-ROUTINE
170 GOSUB 390
180 PRINT
190 REM WER GEWANN?
200 IF M=C THEN LET S$ = "UNENTSCHEIDEN"
210 IF M>C THEN LET S$ = "SIE HABEN GEWONNEN"
220 IF M<C THEN LET S$ = "COMPUTER HAT GEWONNEN"
230 REM PRINT ERGEBNIS
240 PRINT S$
250 END
260 REM
270 REM
280 REM
290 REM
300 REM SUBROUTINE 'WUERFELN'
310 REM
320 LET D=INT(10*RND)
330 IF D>6 THEN GOTO 320
340 IF D<1 THEN GOTO 320
350 RETURN
360 REM
370 REM
380 REM
390 REM
400 REM
410 IF D=1 THEN GOSUB 530
420 IF D=2 THEN GOSUB 600
430 IF D=3 THEN GOSUB 670
440 IF D=4 THEN GOSUB 740
450 IF D=5 THEN GOSUB 810
460 IF D=6 THEN GOSUB 880
470 RETURN
480 REM
490 REM
500 REM
510 GRAFIK-SUBROUTINEN
520 REM
530 PRINT " "
540 PRINT " "
550 PRINT " "
560 PRINT " "
570 PRINT " "
580 PRINT " "
590 RETURN
600 PRINT " "
610 PRINT " "
620 PRINT " "
630 PRINT " "
640 PRINT " "
650 PRINT " "
660 RETURN
670 PRINT " "
680 PRINT " "
690 PRINT " "
700 PRINT " "
710 PRINT " "
720 PRINT " "
730 RETURN
740 PRINT " "
750 PRINT " "
760 PRINT " "
770 PRINT " "
780 PRINT " "
790 PRINT " "
800 RETURN
810 PRINT " "
820 PRINT " "
830 PRINT " "
840 PRINT " "
850 PRINT " "
860 PRINT " "
870 RETURN
880 PRINT " "
890 PRINT " "
900 PRINT " "
910 PRINT " "
920 PRINT " "
930 PRINT " "
940 RETURN

```


gleich ausfallen. Innerhalb dieser Zeit wird der Wert von R wahrscheinlich auf einige Hundert angestiegen sein.

Die Division zum Reduzieren des Wertes von R auf eine Zahl unter 10 findet nicht statt, solange nicht die Leertaste gedrückt ist. Dies bedeutet, daß R sich meist im Bereich von einigen Hundert bewegt, ehe die Division stattfindet. Die Lösung dieser zeitlichen Verzögerung besteht darin, den Test „größer als die obere Grenze“ zum Teil der ersten Schleife zu machen. Dazu das folgende Programm:

```
5 LET R=0
10 PRINT "DRUECKEN SIE DIE LEERTASTE"
20 FOR X=0 TO 1
30 LET R=R+1
40 IF R>9 THEN LET R=1
50 IF INKEY$="" THEN GOTO 80
60 LET X=X+1
70 NEXT X
80 PRINT R
```

In diesem Programm kann R niemals kleiner sein als 1 oder größer als 9. In der Zeit, die verstreicht, bis die Leertaste gedrückt (und von der INKEY\$-Funktion in Zeile 50 erfaßt) wird, beinhaltet R einen Wert, der im Bereich von 1 bis 9 liegt.

BASIC-Dialekte

RANDOMIZE

Beim BBC Micro und dem Oric Zeile 20 löschen und Zeile 320 ersetzen durch:
320 LET D=INT(10*RND(1))

RND

Beim Dragon Zeile 20 löschen und Zeile 320 ersetzen durch:
320 LET D=RND(6)
und Zeilen 330 und 340 löschen.

Beim VC-20 und Commodore 64 Zeile 20 ersetzen durch:
20 LET X=RND(-1)
und Zeile 320 ersetzen durch:
320 LET D=INT(10*RND(1))

INKEY\$

Beim VC-20 und dem Commodore 64 Zeile 40 beim ersten Listing ersetzen durch:
40 GET A\$
und Zeile 50 beim kompletten Listing ersetzen durch:
50 GET A\$: IF A\$="" THEN GOTO 80
Beim ZX81 Zeilen 10 und 50 modifizieren, indem die Leerstelle (" ") ersetzt wird durch "X".

Beim BBC Micro ist INKEY\$(10) zu ersetzen. Die in Klammern stehende Zahl ist die Zeit in Hundertstelsekunden, die das System auf einen Tastendruck wartet.

Übungen

RND-Funktion

Modifizieren Sie das letzte Programm so, daß Sie eine Zufallszahl im Bereich von 1 bis 6 erhalten.

Schleife und Durchschnittszahl

Erweitern Sie das letzte Programm so, daß es 100mal durchläuft und die Durchschnittszahl der 100 Ergebnisse liefert.

INKEY\$

Wie würden Sie unter Verwendung der INKEY\$-Funktion ein Programm schreiben, das irgendein auf der Tastatur eingegebenes Zeichen liest und den Ausdruck liefert: DIE TASTE, DIE SIE GEDRUECKT HABEN, WAR:*. (* repräsentiert die Taste, die gedrückt wurde)?

Taktschleife

Schreiben Sie eine Taktschleife (Zählschleife), und verwenden Sie die INKEY\$-Funktion, um herauszufinden, wie groß der Wert einer Variablen nach 10 Sekunden ist. Schreiben Sie das Programm so, daß der letzte Ausdruck lautet: DER WERT VON R IST NACH 10 SEKUNDEN:*. (* repräsentiert den Wert von R).

IF-THEN-Test

Schreiben Sie ein einfaches Spielprogramm, in dem der Computer eine im Bereich von 1 und 100 liegende Zufallszahl generiert und der Spieler raten muß, welche Zahl es ist. Der Spieler hat fünf Versuche frei. Nach jedem Versuch antwortet das Programm mit einer der folgenden Mitteilungen: SIE LIEGEN ZU HOCH, SIE LIEGEN ZU NIEDRIG, SIE HABEN RICHTIG GERATEN, GRATULIERE oder KEIN WEITERER VERSUCH, SIE HABEN VERLOREN!

Lösungen in der nächsten Folge.

Lösungen zu den Übungen vom letzten Mal

Schleife 1

DER WERT VON A IST 450

Schleife 2

START
STOP

Schleife 3

DER WERT VON A IST JETZT 160

Schleife 4

SCHLEIFEN OHNE ENDE
SCHLEIFEN OHNE ENDE
SCHLEIFEN OHNE ENDE
bis zur Programmunterbrechung durch RESET oder BREAK.

READ-Übung 1

ICH TESTE DIE READ-ANWEISUNG.
170

READ-Übung 2

X=1
X=2
:
:
:
X=23



Computern in der dritten Dimension

3-D-Computerspiele sind eine interessante und abwechslungsreiche Variante einer 30 Jahre alten Technik, die schon in Comics und Kinofilmen Furore machte.

Vor einigen Jahren startete die ARD eine 3-D-Fernsehsendung zu Testzwecken. Der durchschlagende Erfolg blieb aus, und es ist unwahrscheinlich, daß dreidimensionale Fernsehbilder, die erst durch eine unförmige Pappbrille betrachtet plastisch wirken, sich jemals durchsetzen werden.

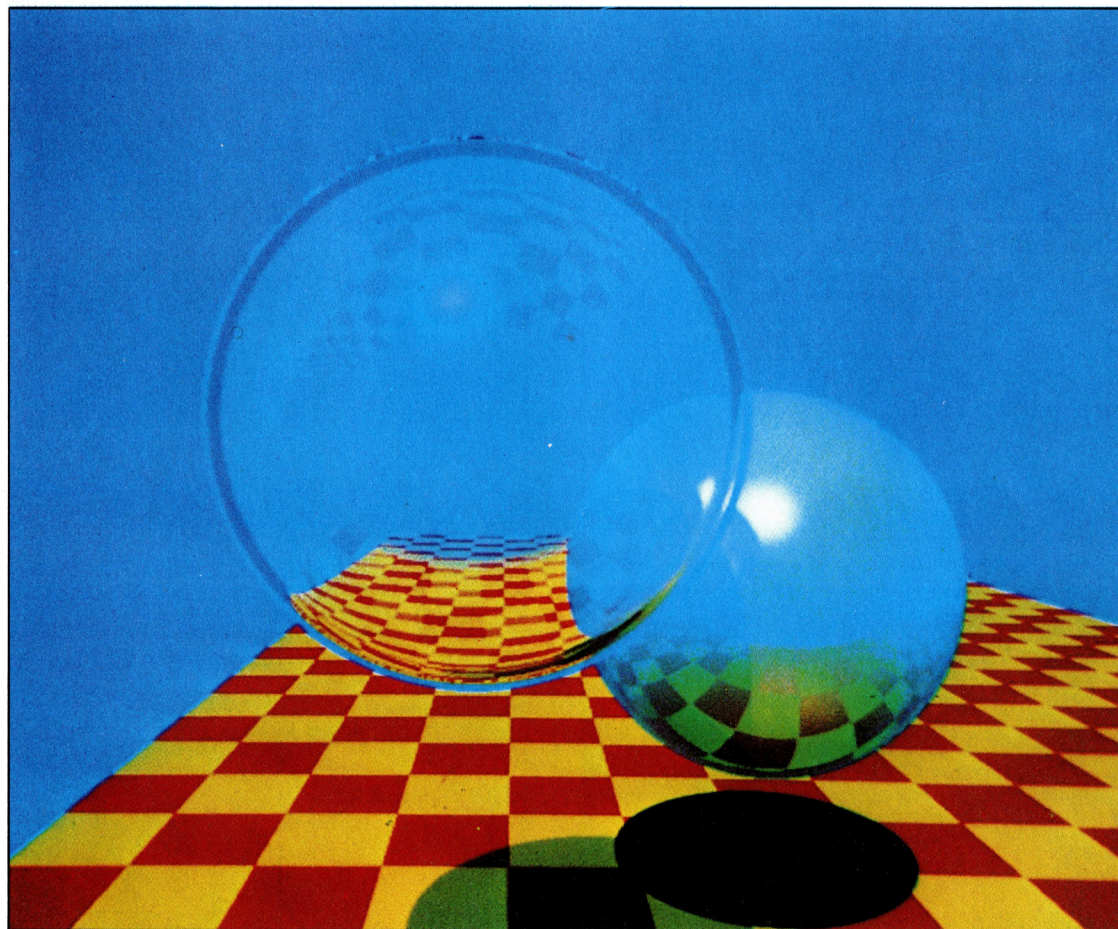
Anders bei Heimcomputern: Zwar sind die Darstellungen lediglich einfarbig, und es lassen sich keine Flächen abbilden; der 3-D-Effekt ist jedoch deutlich wahrzunehmen. Einige kommerzielle Programme schmücken sich zu unrecht mit der Eigenschaft „dreidimensional“. Meist handelt es sich in solchen Fällen um gewöhnliche Computerspiele ohne Tiefenwirkung, bei denen mit zusätzlichen Schattierungen und anderen Tricks der 3-D-

Effekt imitiert werden soll.

Echte 3-D-Spiele zeichnen sich durch Bewegung in der dritten Dimension aus. Die dargestellten Objekte müssen sich zum Betrachter hin und von ihm weg bewegen können. Aber auch hier ist ohne die spezielle 3-D-Brille dieses Vergnügen leider nicht zu genießen. Die Brillengläser sind farblich verschieden; eines ist rot, das andere grünblau (cyan).

Falls Sie die ARD-Versuchssendung nicht gesehen haben, so erinnern sie sich sicher an Darstellungen in Geometrie- und Bilderbüchern, oder Sie kennen Kinofilme, die erst durch eine solche Brille betrachtet räumlich wurden. Man kann sehr leicht feststellen, ob es sich um echte oder falsche 3-D-Bilder handelt. Die richtigen dreidimensionalen Bilder beste-

Der Künstler und Computerwissenschaftler Turner Whitted demonstriert mit diesem Bild Spiegelungen. Beim Drehen fangen die Kugeln die Spiegelung des Hintergrundes ein und geben sie an andere Kugeln weiter. Der Computer schafft sozusagen eine zusätzliche Dimension.





hen aus sich scheinbar sinnlos kreuzenden grünblauen und roten Strichen, die ohne Brille betrachtet einen recht verwirrenden Eindruck machen.

Bildschirm-Darstellungen, die mehrere Farben enthalten und ohne Schwierigkeiten sinnvolle Formen erkennen lassen, sind mit Sicherheit nicht dreidimensional. Computerspiele dieser Art haben keine wirkliche Tiefe, sie verfügen nicht über die dritte Dimension; die Bewegungen der Objekte finden also nur in zwei Richtungen statt – von oben nach unten und von rechts nach links bzw. umgekehrt.

Die Erzeugung von Bildern, die beim Betrachter eine räumliche Wahrnehmung hervorrufen, ist ziemlich einfach. Die für diesen Effekt notwendigen mathematischen Berechnungen sind jedoch recht verwirrend, weil der Blickwinkel der Augen unterschiedlich ist. Von jeder Szene und von jedem Objekt müssen also zwei Darstellungen mit entsprechend abweichender Perspektive angefertigt werden, so als würde ein Zeichner von zwei verschiedenen Positionen aus arbeiten. Um die Bilder voneinander unterscheiden zu können, wird eines rot und das Gegenstück grünblau gezeichnet.

Wenn man ein 3-D-Bild plastisch sehen will, muß man die bereits erwähnte Brille aufsetzen. Ist das rechte Glas grünblau gefärbt, so wird man mit dem rechten Auge lediglich die roten Linien sehen. Die andere Farbe wird herausgefiltert. Das linke Brillenglas muß dementsprechend die grünblauen Linien durchlassen, folglich muß es rot sein. Auf diese Weise lassen sich beide Darstellungen exakt trennen. Das Resultat wird vom Gehirn als dreidimensionales, also räumliches, Hintereinander wahrgenommen.

Farbzuordnung

Die Zuordnung von Brillenglas-Einfärbung und Auge hängt davon ab, welche Farbe für die Striche einer Darstellung gewählt wurde. Wurde für die Betrachtung aus der linken Position die grünblaue Farbe gewählt, muß das linke Brillenglas rot sein.

Der große Nachteil bei diesem Verfahren ist, daß die dreidimensionalen Darstellungen einfarbig wahrgenommen werden. Rote und grünblaue Striche gemischt, werden vom menschlichen Hirn als eine neue, mehr oder weniger schwarz-weiße Strichzeichnung erkannt.

Wie aber erfolgt die Umsetzung, wenn sich ein Objekt zum Betrachter hin oder von ihm weg bewegt? Wir wissen aus Erfahrung, daß Gegenstände mit größerem Abstand kleiner erscheinen. Soll eine Bildschirm-Darstellung echt wirken, so muß der Computer die Objektgröße der Bewegung entsprechend ständig neu berechnen und verändern. Hierzu verwendet der Programmierer die trigonometrischen Funktionen, genauer die Tangensfunktion, de-

ren Name in den meisten BASIC-Dialekten mit TAN angegeben wird.

Die Berechnung der Bildschirm-Koordinaten ist umfangreich und deshalb recht langsam. Sollen die Bewegungen natürlich wirken, dürfen die Darstellungen nicht zu komplex sein: Je schneller sich ein Objekt bewegen muß, desto einfacher muß seine Gestalt sein. Ähnlich verhält es sich mit der Anzahl der abzubildenden Gegenstände. Sind es zu viele, kommt es wegen der langen Rechenzeit zu einem abgehackten, langsamen Bewegungsablauf.

Ein weiteres Problem ist die begrenzte Auflösung der meisten Fernsehbildschirme, denn sie bestimmt maßgeblich, wann ein markantes Objekt zum unkenntlichen Klecks verschwimmt. So gesehen bestimmt die Auflösung auch, wie weit sich ein Objekt vom Betrachter entfernen kann. Sobald die Berechnungen für die linke und die rechte Darstellung die gleichen Werte ergeben, werden die Objekte unscharf oder sogar unkenntlich, und alles weitere Rechnen wäre sinnlos. Um die

Mike Singleton hat Heimcomputer-Spielen eine neue Perspektive gegeben, indem er eine mehr als 30 Jahre alte Technik, die schon in Comic-Heften auftauchte, für Computergrafik anwendbar machte. Bei der Entwicklung von 3-D-Spielen hat er die Brille mit dem roten und dem grünblauen Glas viele Stunden lang getragen. Er glaubt jetzt, daß sein Gehirn den 3-D-Effekt bereits ohne Brille wahrnimmt.



3-D-Wirkung nicht aufzuheben, dürfen sich Bildpunkte der beiden farbigen Darstellungen nicht zu nahe kommen. Sie müssen mindestens einen Punkt voneinander entfernt sein.

Dabei zeigt sich, wie wichtig die Höhe der Auflösung des Bildschirms für gute dreidimensionale Darstellungen ist. Spezielle Computermonitore liefern erheblich bessere Ergebnisse als gewöhnliche Fernsehgeräte. Die Farben werden besser gezeichnet, und die gezeigten Objekte werden genauer dargestellt. Leider gibt es nur wenige hochwertige Programme, bei denen das beschriebene Verfahren angewendet wird.

Die Programme, die auf diese Weise entstehen, zeichnen sich durch natürlich wirkende und besonders unterhaltsame 3-D-Computergrafik aus.



Farb-„DEMO“ und LOGO-Telefonbuch

Mischen von Farbgrafik und Text mit dem DEMO-Programm. Für geordnete Adressen- und Telefonverzeichnisse sorgt ein einfaches, aber effektives Datenprogramm.

Diesmal zeigen wir Ihnen zwei LOGO-Programmbeispiele. Die ersten Prozeduren verwenden die Turtle-Grafik, mit der bewegte Bilder auf dem Bildschirm erzeugt werden. Das zweite Programm ist ein einfaches Beispiel für eine Datenbank, in der man Adressen und Telefonnummern ablegen kann. Beides sind relativ unkomplizierte Programme, und es dürfte Ihnen anhand der bereits erklärten Befehle keine Schwierigkeiten bereiten, den Arbeitsablauf zu analysieren. Zum besseren Verständnis sollten Sie die Prozeduren nicht einfach eintippen, sondern sich überlegen, wie die Programme strukturiert und gegebenenfalls verbessert werden können.

Texte mit Grafik

Das Programm „DEMO“ besteht aus neun einzelnen Prozeduren, wobei die Hauptprozedur die verschiedenen Bildschirmfarben, die Farben der Turtles sowie die Ausgangspositionen der Turtles definiert. Wie Sie sehen, werden in diesem Programm alle vier Turtles, die beim Atari-LOGO darstellbar sind, eingesetzt.

Nachdem die Anweisungen, die den Bildschirm betreffen, abgehandelt sind, wird zunächst zur LOGO-Prozedur verzweigt, wobei das Wort LOGO auf dem Bildschirm erscheint. Hier wird also demonstriert, wie man in LOGO Texte mit Grafik mischen kann. Versuchen Sie doch mal, eigene kombinierte Text-/Grafikprogramme zu schreiben!

Sobald die vier Turtles das Wort LOGO auf den Schirm geschrieben bzw. gezeichnet haben, legt die Prozedur DEMO die neuen Koordinaten der Turtles fest und ruft anschließend die Befehle von E1 und E2 auf. Die nachfolgenden Anweisungen bewirken, daß die Turtles in unterschiedlichen Abständen und mit verschiedenen Richtungen über den Bildschirm wandern. Das Programm wird mit Eingabe des Wortes DEMO gestartet.

```
TO DEMO
  SETBG 2 SETPC 0 24 SETPC 1 32 SETPC 2 70
  TELL [0 1 2 3]
  HT FS CS
```

```
TELL 0 PU SETPOS [-130 30] PD SETC 7
TELL 1 SETPN 2 PU SETPOS [-130 -60] PD
  SETC 24
TELL 2 SETPN 2 PU SETPOS [20 30] PD SETC
  56
TELL 3 SETPN 1 PU SETPOS [20 -60] PD
  SETC 80
TELL [0 1 2 3] LOGO
TELL 1 PU SETPOS [-80 90] ST
TELL 2 PU SETPOS [-40 90] ST
TELL 3 PU SETPOS [40 90] ST
TELL 0
PU HOME RT 90 PD E1
LT 90 E2
PU SETPOS [80 90] ST
WAIT 120
TELL [1 3] RT 45
TELL [0 1 2 3] SETSP 40
END

TO E1
  REPEAT 16 [FD 10 LT 90 FD 10 RT 90 FD 10
    RT 90 FD 10 LT 90]
END

TO E2
  REPEAT 10 [FD 10 LT 90 FD 10 RT 90 FD 10
    RT 90 FD 10 LT 90]
END

TO E4 :N :S :W
  REPEAT :N [FD :S RT :W]
END

TO G :S
  PU FD :S / 3 PD FD :S / 3 E4 16 2 10 PU E4 2
    2 10 FD :S / 3 RT 90 PD FD :S / 3 BK :S / 3
    LT 90 E4 18 2 10
END

TO O :S
  PU FD :S / 3 PD FD :S / 3 E4 18 2 10 FD :S /
    3 E4 18 2 10 HT
END

TO E5 :S :F
  PU RT 90 FD :F * :S LT 90 PD
END
```




```
TO L :S
  HT
  FD :S RT 90
  PU FD :S / 2 RT 90 FD :S RT 90 PD
  FD :S / 2 RT 90
END
```

```
TO LOGO
  L 30 E5 20 1
  O 30 PU BK 10 E5 30 1
  G 30 PU BK 10 E5 30 1
  O 30
END
```

Das zweite Programm stellt ein einfaches Telefonnummern-Verzeichnis dar, mit dem Sie Daten eingeben, verändern und abfragen können. Durch Aufrufen der Prozedur INIT wird das Hauptprogramm gestartet. Gleichzeitig werden die dazugehörigen Routinen aktiviert. Die Prozedur START bewirkt, daß das Hauptmenü auf dem Bildschirm erscheint, wobei man die drei Unterroutinen (ADD, FRAGEN und SPEICHERN) durch Eingabe der Zahlen 1 bis 3 anwählen kann. Die ADD-Prozedur setzt die eingegebenen Daten, bestehend aus dem Vornamen, dem Nachnamen und der Telefonnummer, auf eine Liste. Wird anstatt des Vornamens das Wort „ENDE“ eingetippt, so gilt der Eingabevorgang als abgeschlossen, und das Programm kehrt zur START-Prozedur zurück. Sobald die Dateneingabe beendet ist, werden die Eingaben mit Hilfe der STORE-Prozedur einer Liste zugeordnet.

Die FRAGEN-Prozedur ruft die beiden Prozeduren FIND und SEARCH auf, fragt die Liste nach einer speziellen Eingabe ab und gibt den Vor- und Nachnamen sowie die dazugehörige Telefonnummer aus. Gibt man einen Namen ein, der nicht in der Datenliste existiert, werden statt dessen Fragezeichen ausgegeben. Bei SEARCH handelt es sich um eine recursive Prozedur, die die Listen VNAME und FNAME nach den entsprechenden Namen durchsucht. Dabei werden auch die bereits erklärten Bedingungsabfragen AND und IF eingesetzt.

Speichern der Telefonnummern

Zum Schluß kommen wir zu der STORE-Routine, bei der alle vorhandenen Daten, inklusive des Programms, gespeichert werden. Wenn Sie also das Programm zu einem späteren Zeitpunkt wieder aufrufen, sind alle zuvor eingegebenen Daten noch vorhanden. Mit INIT werden die Eingaben gelöscht.

```
TO ITEM :A :B
  IF :A = 1 [OP FIRST :B]
  OP ITEM :A — 1 BUTFIRST :B
END
TO START
  TS CT
  PR [TELEFON — BUCH HAUPTMENUE]
  REPEAT 3 [PR [ ]]
```

```
PR [1 ... ADD]
PR [2 ... FRAGEN]
PR [3 ... SPEICHERN]
MAKE "WAHL RC
IF NOT NUMBERP :WAHL [START]
IF :WAHL < 1 [START]
IF :WAHL > 3 [START]
RUN ITEM :WAHL :RUNLIST
START
END
TO INIT
  MAKE "RUNLIST [[ADD] [FRAGEN]
    [DISK.STORE]]
  MAKE "VNAME [ ]
  MAKE "FNAME [ ]
  MAKE "T.NUM [ ]
  START
END
TO ADD
  TS CT
  PR [TELEFON — ADD FUNKTION]
  REPEAT 3 [PR [ ]]
  TYPE [VORNAME ... ] MAKE "VOR RL
  IF :VOR = [ENDE] [STOP]
  TYPE [FAMILIENNAME ... ] MAKE "FAM RL
  TYPE [TEL NUMMER ... ] MAKE "TEL RL
  STORE
  ADD
END
TO FRAGEN
  TS CT
  PR [TELEFON — INQUIRE FUNKTION]
  REPEAT 3 [PR [ ]]
  TYPE [VORNAME ... ] MAKE "VOR RL
  TYPE [FAMILIENNAME ... ] MAKE "FAM RL
  FIND
END
TO DISK. STORE
  SAVE "D:TELBUCH
END
TO STORE
  MAKE "VNAME SE :VNAME :VOR
  MAKE "FNAME SE :FNAME :FAM
  MAKE "T.NUM SE :T.NUM :TEL
END
TO FIND
  TS CT
  PR [TELEFON — INQUIRY]
  REPEAT 3 [PR [ ]]
  TYPE [VORNAME ... ] PR :VOR
  TYPE [FAMILIENNAME ... ] PR :FAM
  TYPE [TEL NUMMER ... ] PR :SEARCH
  :VNAME :FNAME :T. NUM :VOR :FAM
  REPEAT 6 [PR [ ]]
  PR [WEITER MIT RETURN]
  MAKE "DUMMY RC
END
TO SEARCH :VLISTE :FLISTE :TLISTE :VN :FN
  IF EMPTY P :VLISTE [OP [????????]]
  IF AND ( FIRST :VLISTE ) = ( FIRST :VN )
    ( FIRST :FNAME ) = ( FIRST :FN ) [OP FIRST
    :TLISTE] [OP SEARCH BF :VLISTE BF
    :FLISTE BF :TLISTE :VN :FN]
END
```

Die bisherigen Teile des LOGO-Kurses waren speziell für Atari-Rechner ausgelegt. Die nächsten Folgen richten sich an Apple- und Commodore 64-Benutzer. Besitzer anderer Heimcomputer können die unterschiedlichen Dialekte im Handbuch nachschlagen.



Gesetze der Logik

Schon ein Jahrhundert vor der Erfindung des Computers veröffentlichte George Boole die von ihm gefundenen Regeln der mathematischen Logik.

George Boole, 1815 in Lincoln als Sohn eines Schusters geboren, glaubte daran, menschliches Denken auf mathematische Weise beschreiben zu können. Er brachte sich selbst Mathematik bei und begann die Erforschung der Logik menschlicher Entscheidungen. Ohne seine revolutionären Ideen wäre die Entwicklung der Computer von heute nicht möglich gewesen. George Boole starb jedoch

schon 1864 – ein Jahrhundert, bevor die rasante Entwicklung der Computertechnologie begann.

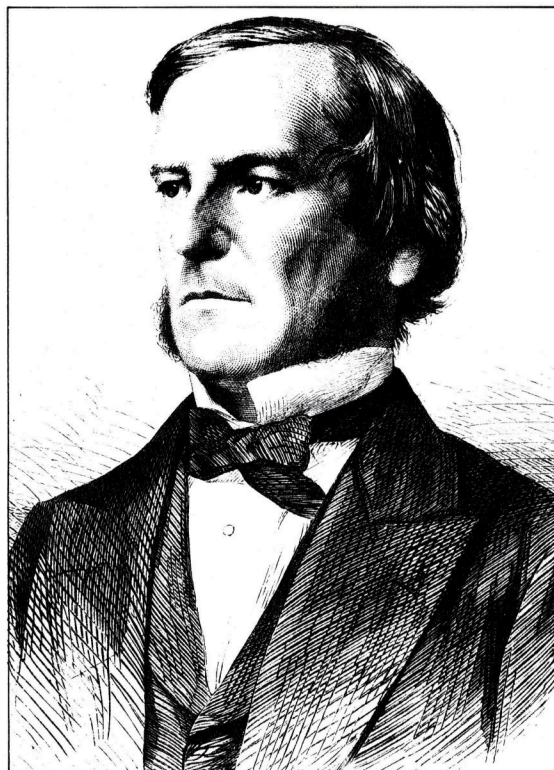
Das Beispiel einer Tanzveranstaltung soll die Denkweise des Mathematikers verdeutlichen. Angenommen, Sie befinden sich in einem Saal und suchen einen Tanzpartner. Einige Paare tanzen, andere Menschen stehen etwas abseits. Logischerweise tut niemand beides gleichzeitig. Ihr auszuwählender Tanzpartner ist entweder männlich oder weiblich, aber nicht beides.

Boole würde die Situation folgendermaßen analysiert haben. Er hätte die Anwesenden vier Gruppen zugeordnet: männliche Personen (m), weibliche Personen (w) und Personen, die gerade tanzen (t) bzw. nicht tanzen (n).

Ein Programm für Freunde

```
10 DIM N$(10), D$(10), F$(10), T$(10), C$(10)
15 REM NAME, TEL.NR., FREUND?, TENNIS?, AUTO?
17 PRINT "GIB DIE DATEN IN DIESER FORM EIN:"
18 PRINT "NAME, TELEFON, JA/NEIN, JA/NEIN, JA/NEIN"
20 FOR K=1 TO 10
30 INPUT N$(K), D$(K), F$(K), T$(K), C$(K)
40 NEXT K
45 REM SUCHROUTINE
50 FOR J=1 TO 10
60 IF F$(J)="JA" AND T$(J)="JA" AND C$(J)="JA" THEN GOSUB 100
70 NEXT J
80 END
100 PRINT N$(J), D$(J)
110 RETURN
```

George Boole lebte in einer Zeit, als man sich Computer noch nicht einmal vorstellen konnte und war doch ein Begründer der mathematischen Logik, die den heutigen Computern zugrunde liegt. 1847 veröffentlichte er seine Erkenntnisse und wurde dadurch fast über Nacht berühmt. Bald wurde er zum Professor für Mathematik an der neugegründeten Universität von Cork.



Informationsstrukturen

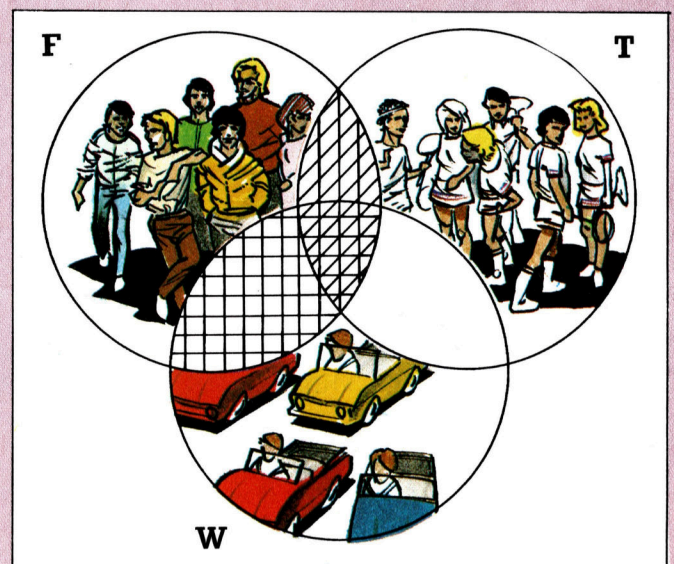
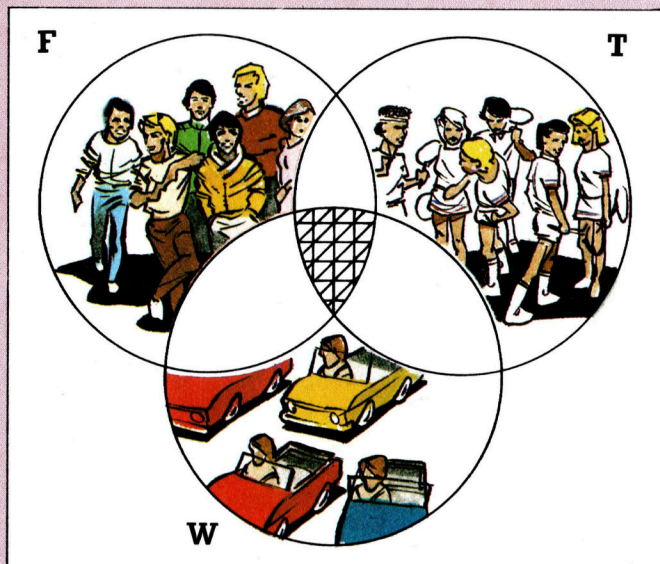
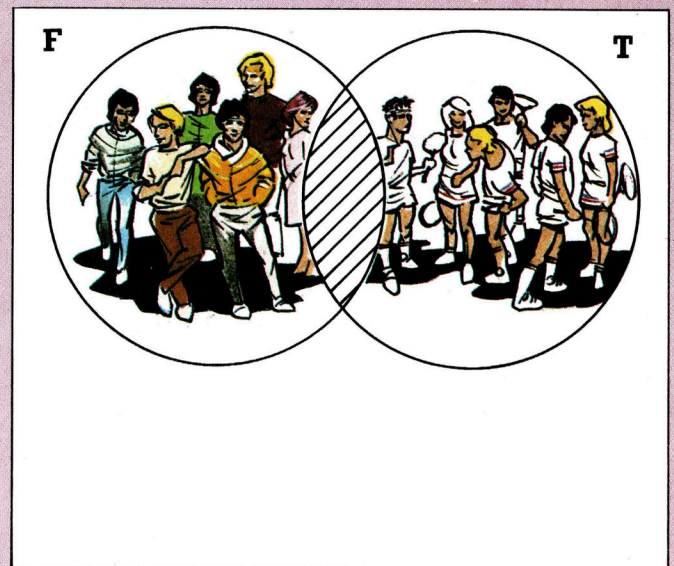
Der Tanzpartner für einen Mann müßte zwei Bedingungen erfüllen: Er müßte weiblich sein und in diesem Augenblick gerade nicht tanzen. Boole erkannte, daß das 'und' bei der Verbindung dieser beiden Bedingungen eine wichtige Funktion erfüllt. Er bezeichnete sie mit einem umgedrehten U. Die Gruppe der möglichen Tanzpartner läßt sich also als $w \cap m$ definieren.

Wenn der Mann nicht tanzen, sondern lediglich eine Unterhaltung führen möchte, so könnte er jeden aus den Gruppen m oder w auswählen, da diese alle im Raum befindlichen Menschen umfassen. Die Funktion 'oder' bezeichnete Boole mit dem Symbol U. Dieser Logik entsprechend beinhaltet $m \cup w$ also alle Männer und Frauen im Raum. Nach diesen Booleschen Symbolen sind die logischen Gatter, mit denen Computer arbeiten, benannt worden.

Nun zu einer interessanten Anwendung der Booleschen Logik. Die englischen Mathematiker John Venn (1834–1923) und Charles Dodgson (1832–1892), besser bekannt als der Schriftsteller Lewis Carroll, entwickelten sie. Als Beispiel soll ein praktisches Problem dienen. Angenommen, Sie haben eine Liste mit den Namen ihrer Freunde und Bekannten im Heimcomputer gespeichert. Zu jedem Namen haben Sie beispielsweise noch die Telefonnummer und das jeweilige Hobby aufgelistet. Eines Nachmittags möchten Sie mit einem Freund oder einer Freundin Tennis auf einem Platz am anderen Ende der Stadt spielen. Ihr



Venn-Diagramme



Ein quadratisches Feld stellt alle Personen dar, die im Computer gespeichert sind. Die Kreise innerhalb des Kastens stellen die individuellen Sets dar. Die Freunde werden in diesem Diagramm der Gruppe F zugeordnet. Die Tennisspieler sind in dem Set T erfasst. Da es einige Personen geben wird, die sowohl Tennis spielen als auch Freunde sind, überlappen sich die beiden Kreise ($F \cap T$). Das Set W umfasst alle Bekannten, die einen Wagen

besitzen. Dieses Set überlappt die ersten beiden. Daher sind die Menschen, die alle drei Bedingungen erfüllen, in der Fläche enthalten, in der sich alle drei Kreise überschneiden ($F \cap T \cap W$). Das letzte Diagramm stellt die gleichen Sets dar (F, T, W); aber die Bedingungen, die erfüllt werden müssen, haben sich verändert. Die neuen Entscheidungskriterien sind, daß die Freunde entweder Tennis spielen oder ein Auto besitzen sollen.

Computer soll die Namen und Telefonnummern aller Personen ausdrucken, die gern Tennis spielen, gute Freunde sind und ein Auto haben (damit sie gemeinsam zum Platz fahren können). Deshalb untersucht das „Programm für Freunde“ (siehe Kasten) zunächst folgende Informationen: Ist es ein Freund/eine Freundin? Hat er/sie einen Wagen? Kann er/sie Tennis spielen?

In dem Programm ist die Anzahl der gespeicherten Personen auf zehn festgelegt. Diese Zahl können Sie selbstverständlich auch ändern. Denken Sie jedoch daran, auch die 10 in der Klammer hinter den DIM-Anweisungen in

Zeile 10 zu ändern. Die Liste wird dann mit einem IF-THEN-Befehl durchsucht, mit dem eine Anzahl von Bedingungen abgefragt werden. Die meisten BASIC-Versionen bieten die Möglichkeit, in dem IF-THEN-Befehl eine ganze Anzahl von Bedingungen mit den logischen Operatoren AND und OR miteinander zu verbinden.

In manchen Programmen gibt es sehr komplexe Kombinationen der logischen Funktionen. Die Bedeutung von Booles Algebra, die zu dessen Lebzeiten nur wenig mehr als ein Kuriosum darstellte, ist erst heute, im Computerzeitalter, erkannt worden.



PEEK und POKE

Mit diesen zwei Befehlen können Sie beim Programmieren all das erreichen, wofür BASIC allein nicht ausreicht.

Das POKEn

Werden POKE-Befehle unüberlegt eingesetzt, kann dies dazu führen, daß das Programm abstürzt. Nachfolgend ein paar POKE-Befehle, die problemlos anzuwenden sind: Bei Atari-Computern bewirkt POKE 751,1, daß der Cursor verschwindet. Geben Sie auf dem Commodore 64 POKE 1024,1 ein. 1024 enthält die erste Adresse des Bildschirms. Nun noch ein Programm für den Sinclair Spectrum:

```
100 FOR N = 0 TO 6
  STEP 2
110 POKE USR "A" + N,
  BINO1010101
120 POKE USR "A" + N
  + 1, BINO10101010
130 NEXT N
140 PRINT "AAAAAAA"
```

Die „A“s in Zeile 140 sind im Grafikmodus einzugeben.

Die Befehle PEEK und POKE werden in BASIC dazu benutzt, Programme zu erstellen, bei denen die Bits und Bytes im Speicher manipuliert werden müssen, um so ein optimales Programmresultat zu erzielen. PEEK wird angewendet, um den Inhalt einer bestimmten Speicherzelle abzufragen; POKE dagegen, um einen Wert (von 0 bis 255) in einer Speicherzelle abzulegen.

PEEK und POKE ermöglichen dem Programmierer den Zugriff auf interne Arbeitsabläufe, was mit einfachen BASIC-Befehlen sehr viel umständlicher oder gar nicht zu bewerkstelligen wäre. Wie bekannt, sind in BASIC bestimmte Speicherplätze reserviert, wo Variablen, Sonderzeichen, Bildschirmpositionen usw. abgelegt werden. Um einfache Programme zu erstellen, muß man nicht unbedingt wissen, wo diese Dinge zu finden sind. Will man jedoch tiefer in die Materie eindringen, bietet PEEK die Möglichkeit, den Inhalt des einzelnen Speicherplatzes abzufragen.

Dieses kurze Programm gibt den Wert einer beliebigen Zelle aus:

```
10 REM ABFRAGEN EINER SPEICHERZELLE
20 PRINT "GEBEN SIE DIE ABZUFRAGENDE
  SPEICHERZELLE EIN."
30 INPUT M
40 P = PEEK (M)
50 PRINT "DIE SPEICHERZELLE ";M;
  " ENTHAELT ";P
60 GOTO 20
70 END
```

Der Inhalt des Speicherplatzes wird als Dezimalzahl dargestellt – obwohl der Computer die Werte intern als Binärzahlen ablegt. Wenn Sie wissen wollen, wie das Äquivalent des momentanen Wertes eines Speicherplatzes auf dem Bildschirm erscheinen würde, fügen Sie in Zeile 50 noch die Funktion CHR\$ ein, wodurch die Dezimalzahl in das entsprechende Zeichen umgewandelt wird.

```
50 PRINT "DIE SPEICHERZELLE ";M;
  " ENTHAELT ";CHR$(P)
```

Mit einer weiteren Programmänderung und der FOR-NEXT-Schleife läßt sich der Inhalt aller abfragbaren Speicherplätze auflisten: In Zeile 20 wird die FOR-NEXT-Schleife mit der Anweisung FOR X = 0 TO 65535 initiiert, Zeile 30 wird um einen Befehl erweitert und in Zeile 60 schließlich die Schleife abgeschlossen.

Damit Sie aber genügend Zeit haben, die dargestellten Werte auch zu lesen, bevor ein neuer Wert ausgegeben wird, muß vor dem NEXT X eine Verzögerungsschleife eingebaut werden, beispielsweise mit dem Zeitfaktor 1 TO 200. Das folgende Programmbeispiel zeigt, wie das Listing für die Abfrage eines 64K-Speichers komplett aussieht. Ist ihr Computer mit einer geringeren Kapazität ausgerüstet, müssen Sie den Endwert der FOR-NEXT-Schleife in Zeile 20 verändern: 16383 für 16 K, 32767 für 32 K oder 49151 für 48 K.

```
10 REM ABFRAGE UND AUSGABE ALLER
  SPEICHERZELLEN
20 FOR X = 0 TO 65535
30 LET Y = PEEK (X)
40 PRINT "SPEICHERZELLE ";X;" = ";Y;" = ";
50 PRINT CHR$(Y)
60 FOR D = 1 TO 200
70 NEXT D
80 NEXT X
90 END
```

Umwandlung in Zeichen

Wie Sie feststellen können, werden sämtliche Dezimalzahlen in ein entsprechendes Zeichen umgewandelt, obwohl nur Werte zwischen 32 und 127 tatsächlich ausdrückbare Zeichen repräsentieren. Bei den meisten Computern sind die Zahlen von 128 bis 255 (das ist die größte Zahl, die ein Byte beinhalten kann) für spezielle Grafikzeichen und Werte zwischen Null und 32 für die Bildschirmausgabe reserviert. Werden diese Kontrollwerte unachtsam in einem Programm eingesetzt, kann dies zu unerwünschten Reaktionen, wie etwa Bildschirmlöschen oder Versetzen des Cursors, führen.

POKE bewirkt genau das Gegenteil von PEEK. Dieser Befehl erlaubt, daß ein Wert (zwischen 0 und 255) in eine Speicherzelle geschrieben wird. Bevor sie POKE benutzen, sollten Sie genau überlegen, welchen Speicherplatz Sie ansprechen und welchen Wert Sie darin ablegen wollen. Wird in eine Speicherzelle gePOKEd, die nicht für das Neudefinieren zugelassen ist, kann dies leicht dazu führen, daß das Programm „abstürzt“. In den meisten Fällen läßt sich dieser Fehler nur durch Drücken der RESET-Taste bzw. durch Abschalten des Computers beheben. Sehen Sie deshalb vor Anwendung der POKE-Befehle im Handbuch Ihres Computers nach, welche POKE-Adressen für den Benutzer frei sind.

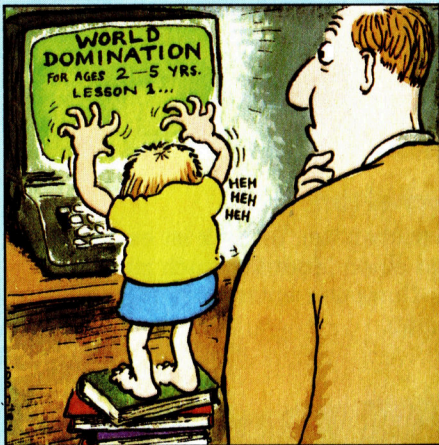
Fachwörter auf einen Blick

Abruf

(fetch) Vorgang, bei dem Daten aus einem Speicher geholt werden

ADA

Höhere Programmiersprache für Echtzeit-Anwendungen; ADA wurde 1980 in den USA entwickelt



ALGOL

Problemorientierte höhere Programmiersprache für Aufgaben im technisch-wissenschaftlichen Bereich; sie wurde Anfang der 60er Jahre entwickelt; ALGOL ist die Abkürzung für ALGOritmic Language (algorithmische Sprache)

Algorithmus

Bezeichnung für einen Rechengang, der genau nach einem bestimmten Schema abläuft; algorithmische Computersprachen wie FORTRAN, COBOL und ALGOL beschreiben die Rechengänge, die mit Hilfe von Algorithmen möglich sind

alphanumerisch

Ein alphanumerischer Zeichensatz besteht aus Buchstaben, Ziffern und Sonderzeichen

Anschluß

(port) Anschlußstelle von Peripheriegeräten an das Bus-System des Mikroprozessors

ANSI-Norm

Technische amerikanische Norm, ähnlich der DIN-Norm in Deutschland; das American National Standards Institute entwickelt und veröffentlicht die ANSI-Normen

CAL

(computer assisted learning) computerunterstütztes Lernen; Einsatz des Computers im Schul- und Lehrbereich, um das Lernen zu rationalisieren, Antworten zu überprüfen und dem Lernenden Hinweise zu geben

COBOL

Abkürzung für Common Business Oriented Language (kaufmännisch orientierte Programmiersprache)

Cartridge

Speichermedium mit serielltem Zugriff

Digit

Ziffer, Zeichen, Stelle

Digitiser

Grafiktablett; es setzt geometrische Positionen in digitalisierte Informationen für den Computer um

DNA (DNS)

Desoxiribonuclein-Acid(-Säure); Träger der Erbinformationen in den Genen von Pflanze, Tier und Mensch

Index-Datei

Dies ist eine spezielle Dateistruktur, bei der den verschiedenen separaten Dateien Indexdateien zugeordnet sind; eine indizierte Datei versetzt den Computer in die Lage, genau festzustellen, an welcher Stelle auf dem Datenträger sich bestimmte Daten befinden, zum schnellen Abruf

Megabyte

1 Megabyte entspricht 1 048 576 (=2²⁰) Byte

Modulator

Er wandelt elektrische Signale um, die gesendet werden sollen

RPNL

(Reverse Polish Notation Language) Höhere Programmiersprache, die den Sprachen FORTH und PASCAL ähnelt

Supraleitfähigkeit

1911 von H. Kamerlingh Onnes entdeckte physikalische Erscheinung: Bei starker Abkühlung verschwindet der elektrische Widerstand vieler Metalle vollkommen; die Leitfähigkeit nimmt zu; das Innere des jeweiligen Drahtes ist frei von Magnetfeldern

Verschachtelung

(nesting) Technik beim Programmieren, die ein Programm in Subroutinen (Unterprogramme) unterteilt; von diesen Unterprogrammen werden wiederum Unterprogramme aufgerufen

Verschlüsselung

Um Daten auf Datenträgern speichern zu können, werden sie binär verschlüsselt, so daß sie für den Computer verständlich sind

Volt

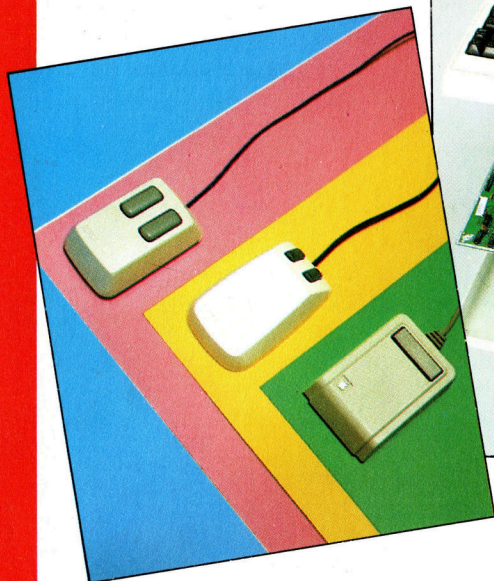
ist die physikalische Einheit, die der elektrischen Spannung entspricht

Bildnachweise:

225: Richard Williams Animation LTD;
227: Ian Dobbie, Walt Disney LTD;
228, 229: Kevin Jones;
232: Chris Stevens;
233: Rolf Seiffe;
235: Bob Freeman;
236: Tony Lodge;
239: Simon Lewis;
240, 241: Liz Dixon;
241: Ian McKinnel;
242: Caroline Holden;
246: m. frdl. Gen. Birkhäuser Verlag;
250: Royal Society;
251: Kevin Jones.

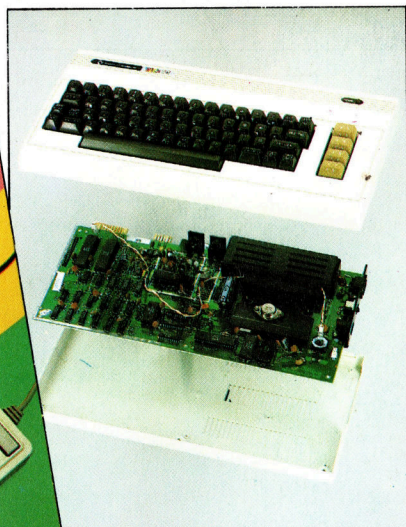
++ Vorschau +++ Vorschau +++ Vorschau +++

computer kurs Heft 10



Mäuse

erleichtern das Auffinden einer Stelle auf dem Bildschirm wesentlich. Außerdem kann die Maus auf dem Grafikbildschirm Striche ziehen und Flächen „anstreichen“.

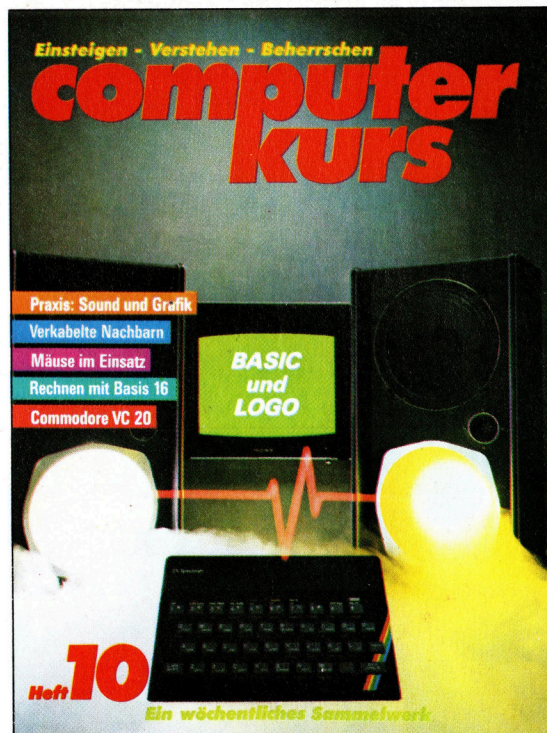
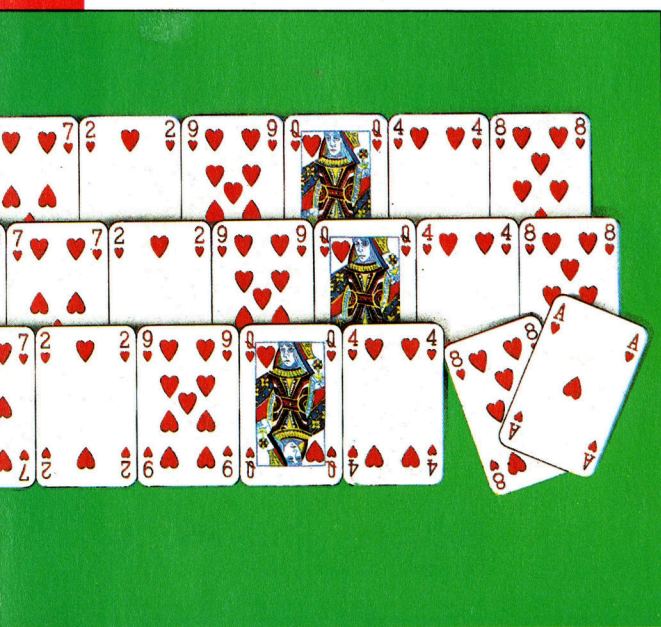


Möglichkeiten

Obwohl er zu den preisgünstigen Geräten gehört, bietet der Commodore VC 20 interessante Grafik-Möglichkeiten: Es sind bis zu 16 Farben verfügbar.

Methode

Daten sortieren muß jeder Computer. Wir stellen ein Sortiersystem vor.



Modellverhalten

In vielen Bereichen der Technik ist es von Vorteil, zuerst einen Modellversuch zu machen. Der Computer untersucht die möglichen Varianten.



+++ Daten im „Wartezimmer“ +++ Sounds
und Grafik aus dem Rechner +++ Unsere
verkabelten Nachbarn +++ Rechnen mit
Basis 16 +++ Die CPU +++ Basic: Neue
Dimensionen +++ Logo-Kurs +++