

Willkommen im Land des Abenteuers

Abenteuerspiele sind eine fantastische Sache. Nicht nur für Spieler, sondern auch für Programmierer eröffnet sich ein weites Feld für kreative Schöpfungen. Wir möchten Ihnen das notwendige Rüstzeug dazu geben.

Lieber Adventure-Freund, auf den folgenden Seiten möchte ich Ihnen gerne erläutern, worum es in diesem Kurs geht – wo Schwerpunkte gesetzt werden, was Sie Neues lernen werden etc. Einige von Ihnen werden sich vielleicht schon mit dem Abenteuerspiele-Programmierkurs, der im Rahmen des 64'er-Sonderheftes 2/85 veröffentlicht wurde, beschäftigt haben. Wer will, kann den folgenden Kurs als eine Fortsetzung beziehungsweise Ergänzung des ersten Kurses betrachten. Lassen Sie mich noch einmal kurz zusammenfassen, worum es im ersten Programmierkurs ging:

Im ersten Abschnitt wurde erklärt, was Adventures sind, wie man sich diese Spiele ausdenkt, sich auf die Programmierung derselben vorbereitet und schließlich, wie man Spielpläne erstellt.

Im folgenden Abschnitt wurde dann ausführlich beschrieben, wie man komfortable Befehlseingabe-, Codier- und Analyseprogramme schreibt. So entstand schließlich ein Programmmodul, das in der Lage war, eingegebene Befehlssätze wie zum Beispiel »Nimm das Schwert und geh nach Norden« in einzelne Worte zu zerlegen, um es anschließend in VERB-, OBJEKT-, GEGENSTANDS- und PERSONENZahl zu codieren.

Im zweiten Hauptteil des Kurses ging es dann um die Actionprogrammierung in Abenteuerspielen – also wie man Spielhandlungen programmiert.

Am Ende dieses Kurses war schließlich jeder Teilnehmer (hoffe ich doch!) in der Lage, Adventures mit den folgenden Eigenschaften zu programmieren:

- Verständnis von ganzen deutschen Sätzen
- Realtime (computergesteuerte Personen wie zum Beispiel Monster im Spiel)
- Programmieren der Adventures ausschließlich mit 38 KByte Basic-RAM
- Speichermedium: Datasette ausreichend.

Kurzum, mit dem anhand des Kurses erworbenen Wissen konnte man die Art von Adventure mit relativ gutem Sprachverständnis (Parser) programmieren.

Wozu also nun wieder ein Adventure-Programmierkurs?

Um es gleich auf einen Nenner zu bringen: Die damals erlernten Programmiertechniken entsprechen nicht mehr dem neuesten Stand derer, die heute unter Einsatz von Floppy-Laufwerken etc. möglich sind.

Während man den ersten Kurs noch komplett mit C 64 und Datasette bewältigen konnte, ist es nun unbedingt erforderlich, mit einem Floppy-Laufwerk zu arbeiten. Man beachte lediglich die Tatsache, daß heuer fast keine professionellen Adventures mehr auf Kassette veröffentlicht werden. Dies ist nicht nur deshalb der Fall, weil die Datasette zu langsam ist,

sondern vielmehr darin zu begründen, daß die Datenmengen, die für ein Super-Adventure benötigt werden, nur noch mit Hilfe eines Disketten-Laufwerks zu bewältigen sind. Beachten Sie doch einmal die folgende Tatsache:

Ein C 64 mit Datasette verfügt im Endeffekt über 64 KByte RAM, die man ordentlich verwalten kann (sequentielle Dateien auf Band sind für die Programmierung von großen Adventures unbrauchbar).

Viel Speicher für Adventures

Ein C 64 mit Floppy-Laufwerk im Gegensatz kann mit den 64 KByte RAM des C 64 und einer Disk-Seite, die eine Kapazität von zirka 170 KByte hat, einen Datenbereich von 234 KByte verwalten. Schon heute gibt es allerdings Adventures wie zum Beispiel Dragonworld, Fahrenheit 451 und Amazon, die auf 4 Diskseiten gespeichert sind und somit einen Umfang von 4 mal 170 KByte – also 680 KByte haben. Diese riesige Speicherkapazität kommt letztendlich durch die zahlreichen Grafiken zustande, die diese Adventures der Firma Telarium aufweisen. Würde man auf Grafik verzichten (à la Infocom), hätte man 680 KByte für Text zur Verfügung. 680 KByte – das sind nicht mehr und nicht weniger als 200 vollgeschriebene A4-Schreibmaschinenseiten! Eine Textfülle dieser Art ist bisher jedoch noch nie bei einem Adventure aufgetreten. Werfen wir doch einmal einen kurzen Blick auf die Infocom-Textadventures. Diese Spiele sind lediglich auf einer Disk-Seite abgespeichert, das heißt, die Spiele verfügen über maximal 60 Seiten Text (wobei ein Filecompactor schon mitberechnet ist). Wenn es nun doch schon möglich ist – warum schreibt denn niemand Super-Adventures mit unglaublichem Textumfang? Ganz einfach – Adventure-Spiele entwerfen und gestalten ist viel Arbeit. Kaum ein Programmierer würde ein Infocom-ähnliches Text-Adventure aus reinem Enthusiasmus beziehungsweise »just for fun« programmieren, da die dafür benötigte Arbeitszeit nicht zu bezahlen wäre – oder würden Sie vielleicht 500 Mark für ein Adventure ausgeben, das den 5fachen Textumfang eines 100-Mark-Programms hat und Sie auch 5mal so viel Zeit zur Lösung kosten würde?

Vielleicht ja, aber der Programmierer muß auch die Tatsache berücksichtigen, daß sein Spiel wahrscheinlich schon wenige Tage nach seinem Erscheinen geknackt wird und somit illegal auf den Markt kommt. Auch ist es sehr schwer abzuschätzen, ob ein Super-Adventure, an dem man etwa ein Jahr lang arbeitet, nach seiner Fertigstellung immer noch etwas Besonderes darstellt, beziehungsweise ob es dann noch aktuell ist. Kurz gesagt, sogenannte Super-Adventures mit riesigem Textumfang, deren Realisierung zwar schon möglich ist, wird es wohl auch in absehbarer Zeit nicht geben. Ein gutes Adventure, das heute auf den Markt kommt, sollte folgende Eigenschaften aufweisen können:

- Guter Textumfang.
- Eventuell ausgezeichnete HiRes-Grafiken.
- Ein Parser, der über ein Vokabular von über 1000 Wörter verfügt und auf jedes Spielkommando intelligent reagiert – also nie einfach nur stupide Antworten gibt wie zum Beispiel »Das geht nicht« auf ein Kommando wie etwa »Nimm das Zimmer«, sondern vielmehr auch begründet, warum dies nicht geht – also zum Beispiel »Sie befinden sich selbst im Zimmer und können es somit auch nicht nehmen«.
- Das Spiel sollte einen so großen Umfang haben, daß ein durchschnittlicher Abenteuerfan zirka 30–40 intensive Stunden zum Lösen des Spiels benötigt.

Ein neuer Trend ist es, Bücher in Adventures umzusetzen. Ein solches Adventure braucht nicht unbedingt viel knifflige Rätsel enthalten, die den Spieler in Verzweiflung bringen. Es geht im Prinzip nur darum, ein Buch in einem neuen Medium

– also dem Computer – zu präsentieren. Ein Spielen dieses Adventures kann somit gleich dem Lesen des Buches gesetzt werden – mit der Ausnahme jedoch, daß sich der Spieler frei »im Buch« bewegen kann und somit noch mehr mit in die Handlung der Geschichte einbezogen wird. Solche Spiele werden dann auch meistens unter Mitarbeit der entsprechenden Buchautoren in Abenteuerspiele umgesetzt.

Zwei erwähnenswerte Spiele dieser Art sind Infocom's »Hitchhikers Guide to The Galaxy« sowie Telarium's »Fahrenheit 451«. Besonders lobenswert ist die Umsetzung von Douglas Adams »Hitchhikers Guide«. Wer die Bücher gelesen hat und dann das Spiel spielt, wird viele identische Textstellen des Buches wiederfinden. Trotzdem hilft es einem im Spiel kaum weiter, wenn man die Bücher kennt, denn hier werden witzige Aufgaben gestellt, die zwar voll und ganz dem Buchstil entsprechen, aber trotzdem nur durch viel Knobeln gelöst werden können. Ein Beispiel dafür ist zum Beispiel eine sprechende Tür in einem Raumschiff. Ich stehe vor der Tür und gebe als Kommando »open door« ein. Die Tür jedoch antwortet ganz frech, daß sie nur demjenigen Durchlaß gewährt, der seine Intelligenz beweisen könnte. Mittels »Inventory« erfahre ich schließlich, daß ich neben vielem Krimskrums und dem Bademantel, mit dem ich bekleidet bin, auch noch eine Zahnbürste bei mir habe.

Gelangweilt gebe ich den meiner Ansicht nach unsinnigen Kommentar »Show door the toothbrush« ein, ohne zu hoffen, daß dieser Kommentar vom Programm verstanden wird. Zur Verblüffung jedoch teilt die Tür mir freundlich mit: »That's only a normal toothbrush – everyone can have that!«. Mehr zu den Infocom-Spielen später.

Wenn Sie sich für die Programmierung dieser neuen Adventure-Generation interessieren, dann ist der nun folgende Kurs genau das Richtige für Sie.

Benötigte Grundlagen

Wer diesen Kurs komplett durcharbeiten möchte, benötigt unbedingt einen C 64 oder C 128 mit Disketten-Laufwerk. Im Prinzip ist kein spezielles Grundwissen erforderlich, abgesehen von der Praxis im Umgang mit Basic natürlich.

Der Kurs enthält zwar zahlreiche Maschinenroutinen, die jedoch so gut beschrieben werden, daß jeder sie benützen und in eigenen Programmen anwenden kann. Wer diese Routinen gemäß eigenen Vorstellungen modifizieren will, muß sich allerdings mit Assembler auskennen. Leichter wird sich noch derjenige tun, der sich mit dem Speicheraufbau – also der Speicherlandschaft des C 64 auskennt. Zwar werde ich auch hier kurz auf die Grundlagen eingehen, es ist jedoch sehr empfehlenswert, vorher Kurse aus der Zeitschrift 64'er

wie zum Beispiel den Grafik-Programmierkurs oder »Memory-Map mit Wandervorschlägen« durchgearbeitet zu haben. Wer sich mit Basic im Rahmen von Stringoperationen etc. noch nicht so gut zurechtfindet, dem möchte ich an dieser Stelle noch den entsprechenden Programmierkurs im ersten Abenteuerspiele-Sonderheft empfehlen.

Wie baut sich der folgende Kurs auf? Wie bereits erwähnt, werden drei große Bereiche behandelt: Dateiverwaltung, Programmierung von künstlicher Intelligenz und Adventure-Programmierung für Fortgeschrittene.

Es bietet sich daher an, einmal zu betrachten, in welchem Zusammenhang diese drei Bereiche stehen, beziehungsweise warum gerade diese drei Bereiche für diesen Kurs ausgesucht wurden.

Dateiverwaltung: Wie es der Name schon sagt, geht es hier um die Verwaltung von Dateien – also irgendwelchen Datenmengen. Effektive Dateiverwaltung ist nur mit einem großen Speicher möglich. Dies ist auch der Grund, warum hierzu unbedingt eine Diskettenstation erforderlich ist.

Was hat Dateiverwaltung denn nun mit der Programmierung mit Abenteuerspielen zu tun? Ganz einfach – betrachten wir doch einmal ein Textabenteuerspiel. Der reine Text bezieht sich auf folgende Dinge:

- Text als Raumbeschreibung
- Text als Beschreibung von Objekten
- Text als Beschreibung von Aktion.

Hierzu einige Beispiele:

Raumbeschreibung: Land of shadow

You are in a land of dark shadows and shallow hills, wicht stretch out in all directions. To the west, the land dips sharply.

Objektbeschreibung: A strage grey machine, shaped somewhat like a clothes dryer, is on one side of the room.

Aktionsbeschreibung: A familiar voice calls down to you. »Are you still there?« he bellows with a coarse laugh. »Well, then, grab onto the rope and we'll see what we can do.« The rope drops to within your reach.

Diese Beispiele habe ich – wie es alle Adventure-Freaks sicherlich bemerkt haben – dem Infocom Adventure ZORK III entnommen.

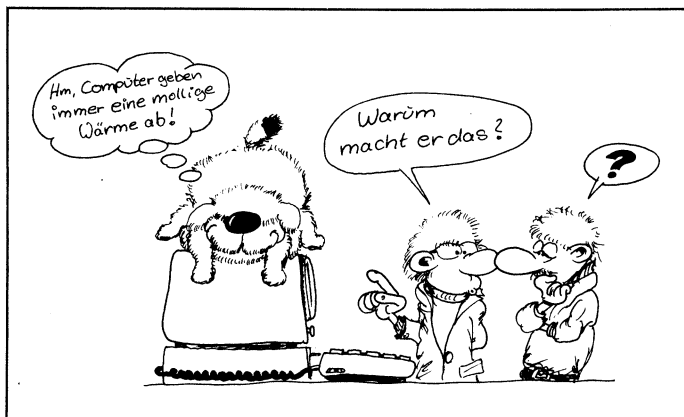
Was ich anhand dieser Beispiele zeigen möchte, ist, daß bei Text-Adventures der gesamte Text in drei Bereichen gegliedert wird, der dann mittels Tabellen verwaltet wird. So gibt es zum Beispiel unter anderem eine Raumbeschreibungstabelle, in der genau steht, an welcher Stelle auf der Diskette (auf der der Text gespeichert ist) sich die Raumbeschreibung zu jedem im Spiel vorkommenden Raum befindet. Angenommen, der Computer benötigt die Beschreibung zum Raum X. Er schaut dann einfach in seiner Tabelle nach, wo sich auf Disk die Beschreibung zu Raum X befindet.

Nun benötigt er noch ein Unterprogramm, das in der Lage ist, den Text von Raum X anhand des von der Tabelle ermittelten Zeiger-Wertes von der Diskette zu lesen.

Ob man nun Adressen, Lagerbestandsteile oder Raumbeschreibungen auf Diskette speichern und verwalten will, spielt keine Rolle – die Grundlagen der Dateiverwaltung sind auf alle Bereiche anwendbar.

Auch im Rahmen der künstlichen Intelligenz (oder abgekürzt KI) ist Dateiverwaltung unbedingt erforderlich.

KI-Programme: Gerade KI-Programme benötigen einen großen Vorrat, aus dem sie ihr »Wissen« beziehungsweise ihren »Einfallsreichtum« schöpfen können. Ich beziehe mich hierbei weniger auf Rechenprogramme als Mastermind-Knacker, sondern vielmehr von ELIZA-artigen Programmen –



als Programme, die in der Lage sind, menschliche Befehle und Fragen zu interpretieren, zu analysieren, um schließlich eine logische Antwort beziehungsweise intelligente »Verlegenheitsaussage« zu geben.

Abenteuerspiele: Auch bei Abenteuerspielen erlangten bei der Programmierung die Dateiverwaltung und die Künstliche Intelligenz immer größer werdende Rollen. Ich spreche hierbei von den sogenannten Super-Adventures, also Adventures mit großem Textumfang und fantastischem Sprachverständnis.

Nun aber zum eigentlichen Kursaufbau.

Kursaufbau

Dateiverwaltung: Hier lernen Sie alle Möglichkeiten der Dateiverwaltungsprogrammierung kennen. Es werden Dateien innerhalb von Basic behandelt, sequentielle Dateien auf Disk und Band, relative Dateien auf Disk sowie Handhabung von index-sequentiellen Dateien. Außerdem erfahren Sie Interessantes über verschiedene Sortieralgorithmen sowie Suchroutinen, die später zum Verwalten des Wortschatzes benötigt werden. Im Rahmen dieses Abschnitts wird auch ein Zeichen-Editor vorgestellt, mit dem wir für die späteren Kapitel einen deutschen Zeichensatz erstellen werden. In einem besonderen Kapitel werde ich auch auf alle Möglichkeiten des Text Traffens zu sprechen kommen – also, wie man Text so kürzt, daß man möglichst viel davon auf einer Disk-Seite unterbringen kann.

Unterstützt wird dieser Kurs-Bereich durch einige kurze, aber äußerst effektvolle Maschinenprogramme, die die Programmierung von Dateiverwaltungen erheblich unterstützen.

Krönung dieses Kursabschnitts ist schließlich ein anwendungsfreundlicher Texteditor, mit dem es möglich ist, Dateien auf Disk zu erstellen (auch speziell in bezug auf Text-Adventures), Dateien auf andere Dateien zu übertragen, Dateien zu sortieren, Dateien auf Drucker auszugeben etc. Dieser Super-Texteditor wird für den ganzen Kurs ein unentbehrliches Hilfsmittel sein.

Adventures: Im Rahmen des Adventure-Kurstteils lernen Sie alles, was man zur Programmierung von Super-Adventures wissen muß. Sie erfahren zunächst, wie man Texte für Adventures ausarbeitet und mittels dem Texteditor auf einer Diskette als relative Datei speichert.

Kern dieses Kursabschnitts ist jedoch zweifellos der Parser 5.0. Mit Hilfe dieses Parsers ist es möglich, Adventures zu schreiben, die mit über 1000 gespeicherten Wörtern nahezu jeden Spielerbefehl verstehen können – also fast den gesamten deutschen Grundwortschatz. Sie erfahren, wie der Parser 5.0 direkt mit dem Texteditor zusammenarbeitet und der Clou: Trotz des unglaublich großen (bei deutschen Text-Adventures bisher zweifellos unerreichten Vokabulars) bleiben dem Programmierer satte 38 KByte Basic-RAM zur absolut freien Verfügung!

Das heißt, jeder kann mit Hilfe des Texteditors und den Parser 5.0 sowie einigen Maschinenroutinen mit einfachen Basic-Kenntnissen Super-Adventures programmieren, wie es sie zur Zeit noch gar nicht gibt!

Damit aber noch nicht genug. Im Rahmen der Adventure-Programmierung wird noch ein anderes kurzes, aber äußerst effektvolles Programm präsentiert: Der Adventure-Cracker – ein Programm, mit dem die Lösung fast jeden Adventures zum reinsten Kinderspiel wird – einfach Adventure-Cracker-Programm starten, Diskette eines Adventures (zum Beispiel Dragonworld, Fahrenheit 451 etc.) einlegen und sich den gesamten Textinhalt des Adventures auf Drucker beziehungsweise Bildschirm ausgeben lassen!

Selbstverständlich lernen Sie gleich anschließend, wie Sie Ihre eigenen Adventures gegen solche Text-Cracker absolut schützen können: Mit Hilfe einer Textverschlüsselung.

Wer glaubt, daß damit genug wäre, irrt – auch die Grafik-Freaks kommen diesmal voll und ganz auf Ihre Kosten. Ich werde ausführlich erläutern, wie Grafik-Bilder der professionellen Spiele entstehen (zum Beispiel die fantastischen Grafiken der Telarium-Adventures). Neben Grundlagen erfahren Sie einiges über Bilder-Digitalisierung etc. Jeder kennt sie – verschiedene Malprogramme wie Koala-Painter und Paint Magic, die zwar alle ihre Vorteile haben, jedoch untereinander nicht kompatibel sind. Auch hier wird mit einigen Übertragungsprogrammen endgültig Abhilfe geschaffen. HiRes-Bilder der bekanntesten Malprogramme in eigenen Adventures einbauen? – kein Problem!

Und zu guter Letzt: Ein Maschinenprogramm, das auf dem Bildschirm ein Text- und ein Grafikfenster darstellt. Anhand einiger Maschinenroutinen können Sie dann zum Beispiel ein Bild mit einem Digitizer erstellen, auf Paint-Magic-Format umwandeln, neu bearbeiten, dann ins Fensterformat umwandeln, um es letztendlich ins eigene Adventure einzubauen.

KI-Programme: Dieser Kursbereich bezieht sich natürlich auf das Programmieren von künstlicher Intelligenz (englisch: artificial intelligence). Es werden Programme erläutert, die diverse Spiele wie etwa NIM spielen können. Dies hat eigentlich nicht mehr sehr viel mit Adventures und Dateiverwaltung zu tun, ist als Einführung zu diesem Thema jedoch nicht wegzudenken. Interessant wird es allerdings, wenn wir zusammen Programme schreiben, die menschliche Befehlseingaben verstehen, interpretieren und darauf antworten können. Wenngleich wir auch erstaunliche Resultate erzielen werden, so sollten Sie Ihre Erwartungen nicht zu hoch stecken! Auch die geschicktesten Programmiermethoden können aus Ihrem C 64 keinen emotionalen Gesprächspartner für einsame Stunden oder lange Winterabende machen. Es wird auch nicht möglich sein, den C 64 innerhalb dieses Kurses im Rahmen der KI total bis ins letzte Bit auszuschöpfen. Um Programme zu schreiben, die den gesamten deutschen Grundwortschatz beherrschen, müßte man zunächst einmal diesen Grundwortschatz in den Computer eingeben. Daß dieser Wortschatz jedoch nicht so einfach in dieser Zeitschrift zum Abtippen gedruckt werden kann, dürfte wohl jedem klar sein. Ich werde Ihnen zwar einige sehr nette Eliza-artige Programme vorstellen, aber wer sich an ganz große Projekte heranwagen will, muß dies mit viel Eigeninitiative tun. Selbstverständlich wird er auch dann im Kurs viele Tips und Tricks, die zum Angehen eines solchen Programms nötig sind, erfahren.

...und vieles andere mehr

Hiermit dürften Sie einen guten Einblick in den Aufbau und Inhalt des Kurses haben. Natürlich werden noch viele Tips und Tricks auftauchen, die ich an dieser Stelle noch nicht erwähnt habe. Wer glaubt, daß dieser Kurs nüchtern und langweilig wird, muß seinen Irrtum spätestens dann zugeben, wenn wir in Kürze folgendes Problem gelöst haben werden:

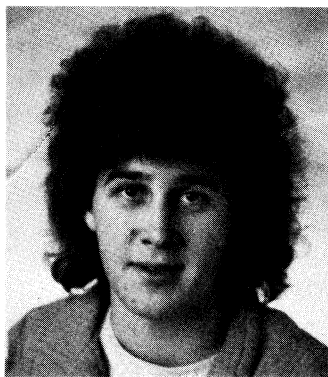
Halten Sie es für möglich, daß in manchen Fällen ein gewöhnlicher Spiegel ein unentbehrliches Hilfsmittel zum Programmieren eines Computers sein kann?

Denken Sie mal nach!

Nun genug der langen Einleiterei. Bitte führen Sie folgende Aktionen in der richtigen Reihenfolge aus:

- Kühlschrank öffnen, Kühlmittel beschaffen (Cola, Bier, etc.)
 - Grundnahrungsmittel bereitstellen (Kartoffelchips)
 - Computer, Floppy, Monitor einschalten
 - Sessel zurechtrücken und gemütlich machen
 - Auf Seite 10 blättern
 - Kurs durcharbeiten
- Los geht's....

(M. Nickles/bs)



Hier mein »Lebenslauf«:

Geboren wurde ich am 18.9.1966 in Erlenbach am Main. Bis zu meinem 10. Lebensjahr besuchte ich den Kindergarten und anschließend die Grundschule. Dann begann die schwärzeste Zeit meines Lebens, der ich in zirka 3 Monaten mit dem Abitur ein Ende bereiten werde: Ich kam ins Gymnasium. Eines der wenigen positiven Dinge, die mir dort begegneten, war ein PET 2001, an dem ich mit 14 Jahren meine ersten Programmierkenntnisse sammelte. Ich war damals ein absoluter Computerfreak und verbrachte die meisten Nachmittage mit diesem PET. Mein Traumcomputer war damals der Atari 800, den ich mir aufgrund des hohen Preises (ca. 2000 Mark) nicht leisten konnte. Drei Jahre später hatte ich endlich genug Geld für einen Homecomputer gespart: natürlich den C 64. Ich stellte schnell fest, daß Computer ein teures Hobby sind, weil man mit der Zeit alle nur

erdenklichen Pheripheriegeräte haben will. Um mit dem C 64 Geld zu verdienen, begann ich Adventures zu schreiben. Nach zahlreichen Wegwerf-Programmen gelang es mir, ein wirklich gutes Adventure zu schreiben: Gordon Saga I. 1984 ging ich mit diesem Programm zur Commodore-Messe in Frankfurt. Bei Happy Software fand ich schnell einen Vertrag für das Spiel. Gleichzeitig lernte ich dort einen 64'er-Redakteur kennen. Ich erzählte ihm von meinen Erfahrungen im Bereich Adventureprogrammierung, und es entstand so letztendlich mein erster Programmierkurs im 64'er-Sonderheft 2/85. Jetzt, ein Jahr später, schrieb ich meinen zweiten Kurs, den Sie im vorliegenden Sonderheft finden.

Trotz meiner aktiven Beschäftigung mit Computern bezeichne ich mich keineswegs als Computerfreak, da ich noch zahlreiche andere Hobbys habe: Ich interessiere mich fürs Filmemachen und bin aktives Mitglied der besten Punk-Gruppe (Musik) der Welt: Brutallica.

Den größten Teil meiner Freizeit verbringe ich im Freundeskreis auf Parties, in Kneipen, Konzerten, im Kino etc. Kälte kann ich nicht ausstehen und fahre bzw. trampe so oft es geht in südliche Länder wie Frankreich und Italien. Mein derzeitiges Idol ist der Programmierer vom Davids Midnight Magic – er hat angeblich schon seinen zweiten Porsche 911...

Ich wünsche Ihnen viel Spaß und Erfolg beim Lesen meines zweiten Programmierkurses, der voraussichtlich das Letzte ist, was ich mit meinem C 64 gemacht habe. Mein nächster Computer ist der Atari 520 ST.

(Michael Nickles)

**Ausführliche Informationen
zu ausgewählten Themen finden
C64-Anwender in zwei
weiteren aktuellen**

64'er

SONDERHEFTEN

SONDERHEFT: C 16, C 116, VC 20 UND PLUS 4

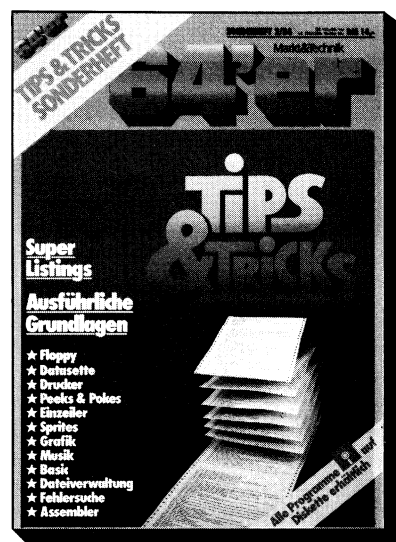
Grundlagen: Fragen und Antworten zum VC 20 und C 16 / Informative Einblicke in Aufbau und Programmierung des VC 20 und C 16 / Allgemeine Vorstellung des C 16-Systems / Daten verwalten mit der Datasette / Maschinensprache für den C 16 und die wichtigsten Interpreter Routinen / Grafik und Sound (C 16) / Hardware: Preiswerte Drucker (C 16) / Grafik: U.a. schnelle Spielegrafik beim C 16 / Grafik-Erweiterung für den VC 20 / Eine Fülle Anwendungs- und Spiel-Listings für VC 20/C 16 / Tips & Tricks: Schnelle Hardcopy (C 16/116) / Ein komfortabler Assembler mit Label (C 16) / So kann man C 64-Programme auf C 16 und VC 20 umschreiben.



**Jetzt für
DM 14,- überall
im Zeitschriften-
handel!**

SONDERHEFT: TIPS & TRICKS

Sound: Drei Stimmen gleichzeitig durch besondere Tastaturabfrage. Grundlagen: So wird Basic schneller / Debugging-Fehlersuche in Basic-Programmen. Grafik: Vom Bit zum Trickfilm / Super Hardcopy bringt den Bildschirminhalt, egal ob Text oder HiRes, auf einen MPS 802. Tips & Tricks-Listings: Disketteninhalt zum Aufkleben / Datasette schneller als Floppy durch Tornado-Tape / Flottes Kopieren mit »Express-Copy« / Filemanager ordnet Disketten und schafft Übersicht / POKES, die man kennen sollte ... und eine Zusammenfassung der besten und nützlichsten Tips & Tricks und Einzeler aus 64'er.



**Nur noch bis
21.04.86
erhältlich!**