

Comal im Brennpunkt

Comal vereint die Stärken der verbreiteten Programmiersprachen Basic, Pascal und Logo in sich. Wann und für wen aber lohnt sich der Einsatz dieser interessanten Sprache auf dem C 64?



Insbesondere Programmier-Anfänger wissen, wie schwer es ist, aus dem C 64 mit einem eigenen Programm anspruchsvolle Grafik, animierte Sprites und dreistimmige Lieder herauszuholen.

Erste Versuche mit dem eingebauten Basic 2.0 trugen eher zur Frustration als zur Motivation bei. Professionelle Spielprogramme aber zeigen, daß der C 64 hierzu durchaus in der Lage ist. Schlußfolgerung: Eine Erweiterung oder eine völlig neue Sprache muß her!

Für den C 64 gibt es inzwischen eine Vielzahl Basic-Erweiterungen, die die Schwächen des Standard-Basics in puncto Grafik, Sprites und Sound wettzumachen versuchen. Comal 80 muß sich — was den Befehlsvorrat und die Geschwindigkeit angeht — hinter keiner verstecken. Die Modulversion (Version 2.01) bietet die Möglichkeit, bis zu 64 KByte ROM und 30 KByte RAM für Programm und Daten zu nutzen. Neben den Standard-Befehlswörtern stehen für die Programmierung:

48 XY-Grafik-Befehle,
49 Turtle-Grafik-Befehle,
22 Sprite-Befehle,
6 Zeichensatz-Befehle,
20 Sound-Befehle und
8 Befehle zur Programmierung von Paddles, Joystick und Lightpen zur Verfügung. Durch die Möglichkeit, selbstdefinierte Prozeduren (also Programmblöcke) in Comal wie Befehle aufzurufen, kann man diesen Befehlssatz in gewisser Weise noch erweitern. Diese Prozeduren brauchen selbst nicht einmal im Programm zu stehen, sondern können sich auf einer Diskette oder auf einem EPROM befinden. Bei Bedarf (Aufruf) werden sie nachgeladen und aktiviert. Auf diese Weise geht kein weiterer Speicherplatz verloren. Maschinensprache-Programmierer können darüber hinaus leicht eigene Spracherweiterungen in Assembler — als Module in EPROMs — als echte Befehlserweiterungen in Comal einbinden. Diese brauchen zur Aktivierung dann lediglich mit USE (Modulname) aufgerufen zu werden. Danach stehen alle neu defi-

nierten Befehle und Funktionen (Erweiterungen) zur Verfügung.

Die Verwaltung relativer Dateien (Random Access Files) wird vom Standard-Basic in keiner Weise unterstützt. In Comal kann man dagegen mit wenigen leicht erlernbaren Befehlen (CREATE, READ FILE, WRITE FILE) die Möglichkeiten der Floppy 1541 komplett nutzen.

Komfortabler Editor

Die Editor-Kommandos von Comal tragen zu einer ausgesprochen komfortablen Programmierung bei. Die Befehle DIR, AUTO, RENUM, DEL(ete) und FIND dürften den meisten Basic-Programmierern nicht unbekannt sein. Besonders nützlich ist weiterhin der Befehl CHANGE, der das Auffinden und (auf Wunsch automatische) Ersetzen von Texten ermöglicht. Die Funktionstasten sind mit nützlichen Kommandos belegt, können jedoch leicht neu definiert werden. Mit dem Befehl DEFKEY (12, "RENUM") wird beispielsweise die Funktionstaste 12 (CBM + F7) mit dem Befehl RENUM belegt. Der Befehl TRACE wird bei Comal, wie auch in einigen bekannten Basic-Erweiterungen, dazu benutzt, nach einem Programmstopp, zum Beispiel bei einem Fehler, die Fehlerquelle festzustellen. Nach der Eingabe von TRACE ist folgende Ausgabe von Comal möglich:

```
the program stopped at
3130 TEXTSTYLE (1,1,0,0)
within
2980 PROC screen
which is called at
1020 screen
within
1000 PROC start
```

Ein weiterer Vorteil — nicht nur für den Programmieranfänger — ist die Fehlerbehandlung. Jede Programmzeile wird unmittelbar nach der Eingabe auf die richtige Syntax geprüft. Die Fehlermeldungen (wahlweise auch in deutsch) sind in vielen Fällen dynamisch, das heißt, sie geben häufig einen Verbesserungsvorschlag. In jedem Fall wird jedoch die fehlerhafte Zeile mit blinkendem Cursor auf dem vom Interpreter vermuteten Fehler ausgegeben. Dadurch wird

die Fehlersuche erheblich vereinfacht. Darüber hinaus kann man mit dem Editor-Befehl SCAN das Programm vor dem Start auf semantische Fehler (zum Beispiel nicht geschlossene Schleifen) untersuchen. Der SCAN-Befehl bietet aber noch mehr: Während der Strukturüberprüfung werden Sprungziele (Labels) in absolute Adressen umgewandelt. Deshalb kann man anschließend jedes als Prozedur definierte Unterprogramm direkt wie einen Befehl aufrufen und starten und somit einzeln auf logische Fehler überprüfen. Eine beispielhafte Eigenart von Comal ist, daß ein Programm vor seinem Start mit RUN automatisch geSCANnt wird. Somit werden Fehler in der Programmstruktur immer rechtzeitig erkannt.

Sehr einfach funktioniert auch das Abfangen von Fehlern innerhalb des Programmlaufs. Die Schlüsselwörter TRAP und ENDTRAP kennzeichnen einen Programmblock in denen Fehler möglich sind (zum Beispiel Ein-/Ausgabefehler). Beim Auftreten eines Fehlers wird in das mit »HANDLER« gekennzeichnete Unterprogramm gesprungen (vergleichbar mit ON ERROR GOTO in Simons Basic).

Strukturiert programmieren

Das wahrscheinlich wichtigste Plus von Comal aber ist die durch diese Sprache unterstützte übersichtliche Programmstruktur. Befehle, die einer strukturierten Programmierung abträglich sind (zum Beispiel GOTO Zeilennummer), sind im Befehlswortschatz gar nicht erst vorhanden. Comal verlangt anstelle einer nichtssagenden Zeilennummer einen frei wählbaren Prozedurnamen (Label). Der scheinbare Nachteil, zu einer Zeile inmitten einer Unteroutine nicht unmittelbar springen zu können, entpuppt sich in der Programmierpraxis sehr schnell als Vorteil: Es ist nämlich mit Comal nahezu unmöglich, unübersichtliche »Spaghetti-Programme« zu erstellen. Der Programmierstil wird zwangsläufig »sauber«. Nachträgliche Ände-

rungen und Erweiterungen des Programms sind deshalb schnell und problemlos durchführbar — auch noch nach vielen Monaten und Jahren. Ein gut strukturiertes Listing erklärt sich auch ohne viele Kommentare quasi von selbst.

Für eine gute Lesbarkeit und Übersichtlichkeit sorgt auch die LIST-Routine des Interpreters. Zeilen innerhalb von Schleifen, Prozeduren und Funktionen werden bei der Ausgabe auf dem Bildschirm oder auf einem Drucker automatisch eingerückt. Weiterhin werden zur besseren Erkennung Befehlskörper in Großbuchstaben, Variablen- und Prozedurnamen dagegen in Kleinbuchstaben ausgegeben. In puncto Schleifen-Vielfalt läßt Comal ebenfalls keine Wünsche offen. Verwendet werden können:

```
IF-THEN-ELIF-ELSE-ENDIF,
CASE-OF-WHEN-
OTHERWISE-ENDCASE,
REPEAT-UNTIL,
WHILE-DO-ENDWHILE,
FOR-TO-STEP-ENDFOR
(NEXT),
LOOP-EXIT-EXIT WHEN-
ENDLOOP.
```

Es dürfte somit unmöglich sein, ein Problem zu finden, das nicht mit einer der möglichen Schleifen optimal gelöst werden könnte.

Von Basic zu Comal

Besonders hervorzuheben ist auch die Funktionsbehandlung von Comal. Im Gegensatz zu Basic können sich nämlich Funktionen, ähnlich den Prozeduren, über mehrere Zeilen hinwegziehen. Ein ganzer Programmblock beliebiger Länge — gekennzeichnet mit FUNC und END-FUNC — kann als Funktion definiert und mit Parameterübergabe vom Hauptprogramm ausgeführt werden. Auch extrem komplexe Funktionen sind auf diese Weise leicht und übersichtlich programmierbar.

Comal ist eine »intelligente« Sprache, die auch den Umstieg von Basic unterstützt. Das Schlüsselwort NEXT beispielsweise ist im Comal-Wortschatz eigentlich nicht enthalten. Wird es dennoch eingegeben, ersetzt der Interpreter diesen Befehl stillschweigend durch

ENDFOR. Weiterhin verlangt Comal genau genommen — an Pascal angelehnt — bei einer Variablen-Zuweisung anstelle des »=« ein »:=«. Aber auch hier ersetzt der Interpreter — im Gegensatz zum Pascal-Compiler — bei Bedarf. Lediglich zwei Dinge sind für den Basic-Programmierer gewöhnungsbedürftig:

- Schlüsselwörter (Befehle) dürfen in Comal nicht abgekürzt werden und
- zwischen den Befehlswörtern und Variablen muß unbedingt ein Leerzeichen stehen. Schreibt man anstelle von »FOR I=« wie gewohnt »FORI=«, interpretiert Comal diesen Ausdruck als Zuweisung der Variablen »FORI« und es kommt unweigerlich zu einer Fehlermeldung.

Die für Comal wichtigen »aktiven Leerzeichen« werden im Programmcode nicht gespeichert und belegen im Gegensatz zu Basic keinen zusätzlichen Speicherplatz.

Comal besitzt nicht nur einen größeren Sprachschatz, sondern ist darüber hinaus auch noch schneller als Basic. Dies wird insbesondere bei Variablen- und Stringoperationen deutlich. Gegenüber Pascal zeichnet sich Comal durch die geringere Strenge und die interaktive Programmierung aus. Es ist deshalb nicht verwunderlich, daß die Verbreitung von Comal fortschreitet. In Schleswig-Holstein ist Comal bereits seit geraumer Zeit offizielle Programmiersprache und Pflicht für den Informatik-Unterricht in den Schulen. Weitere Bundesländer befinden sich zur Zeit noch in der Entscheidungsphase. In den Universitäten findet Comal zunehmenden Anklang und auch privat wird immer häufiger in dieser Sprache programmiert. In vielen Städten und Ländern werden Comal-User-Groups gegründet, die für die Verbreitung der Sprache sorgen und Comal-Anwender mit den neuesten Informationen und nützlichen Comal-Utilities versorgen. Zum Erfolg beigetragen hat sicherlich auch die Free-Soft-Version 0.14, die für den C 64 verfügbar ist. Hiermit ist bereits sehr komfortabel zu

programmieren. Nachteilig ist aber vor allem der geringe Speicherplatz dieser Disketten-Version (9902 Bytes frei), was jedoch durch die Möglichkeit, Programmteile nachzuladen, weniger ins Gewicht fällt. Zum Kennenlernen der Sprache ist sie in jedem Fall empfehlenswert. Wer jedoch alle Möglichkeiten der Grafik-, Sound-, Sprite- und Kontrollportprogrammierung nutzen will, muß mehr Geld investieren. Die Modulversion Comal 80 (Version 2.01) mit 30714 Bytes frei, die von der Comalgruppe Deutschland D.Belz vertrieben wird, kostet inklusive deutschem Handbuch 198 Mark.

Ebenfalls im Preis begriffen ist eine Demodiskette, die eine Reihe interessanter Programme und Prozeduren enthält.

Beeindruckend ist vor allem das Programm »ALLES ZUGLEICH«, bei dem gleichzeitig Grafik-, Sprite- und Sound-Befehle ausgeführt werden. Hierbei wird eine Geschwindigkeit und Perfektion erreicht, die man sonst allenfalls von professionellen Spielprogrammen in Maschinensprache erwarten würde.

Comal-Training

Für die C 64-Modulversion 2.01 bietet die Comalgruppe Deutschland unter anderem auch ein Lernprogramm mit dem Namen »Bruno der Hamster« an (39 Mark), das vor allem in Schulen eingesetzt wird. Es soll den Benutzer allgemein mit strukturierten Programmiersprachen und somit natürlich auch mit Comal vertraut machen. Die Aufgabe bei diesem Programm besteht darin, einen symbolisch dargestellten Hamster auf dem Bildschirm programmgesteuert um Hindernisse zu lenken, Gegenstände (Körner) aufzunehmen und wegzulegen. Erschwert wird dies durch die bewußt eng begrenzte Anzahl der Steuerbefehle (links, vor), so daß man gezwungen ist, eigene Prozeduren zu definieren, um die Aufgabe mit möglichst wenigen Schritten zu lösen. Der Einsatz von Schleifen ist hierbei unverzichtlich. Besonders interessant ist schließlich noch die Möglichkeit, die Landschaft mit einem Editor selbst zu gestalten und somit den Schwierigkeitsgrad der Aufgabe zu bestimmen. (nj)

64'er Comal-Sonderservice

Gelegentlich treffen in unserer Redaktion Anfragen von Lesern ein, warum in der 64'er bislang keine in Comal geschriebenen Programme veröffentlicht werden. Wir werden deshalb in der nahen Zukunft durch eine Leser-Umfrage den Verbreitungsgrad und das Interesse an Informationen über diese Sprache ermitteln. Bereits jetzt möchten wir Ihnen aber die Möglichkeit geben, Comal selbst kennenzulernen. Deshalb haben wir uns entschlossen, die Free-Soft-Version 0.14 auf unserer Programm-Service-Diskette Nr. 8/86, selbstverständlich ohne zusätzliche Kosten, anzubieten und hoffen damit zur weiteren Verbreitung und zum Kennenlernen von Comal beitragen zu können. Sollte das Interesse an dieser Sprache groß genug sein, werden wir dem zukünftig durch mehr Information Rechnung tragen. Die nachfolgend aufgeführten Comal-User-Groups stehen

Ihnen gerne für weitere Informationen über das aktuelle Angebot rund um Comal zur Verfügung.

Comalgruppe
Deutschland
D. Belz
2270 Utersum/Foehr

Comal User Group
Bielefeld
c/o K.D. Polloczek
Paulusstr. 19
4800 Bielefeld 1

Comal Bruger Gruppe
Mindegade 42
8700 Horzens
Denmark

Comal Today DK
Postbox 1222
2300 Copenhagen S
Denmark

Comal User Group
Netherlands
Dick Klingsens
Violeir 3
2925 TE
Krimpen ad IJssel
Niederlande