COMAL for the Commodore 64

other sources of COMAL from http://www.macharsoft.demon.co.uk

C64, COMAL_DOWNLOAD_INDEX

14 COMAL disks

A guide to COMAL

from TRANSACTOR, vol6, #3 - 1985

An Introduction to COMAL: Better than BASIC

Chris Zamara, Technical Editor

An Introduction to COMAL: Better than BASIC

This article is not a product review, but presents information about a product which we feel is significant to the Commodore community.

What is COMAL? If you’re a COMAL fan and drive around with an ‘I speak COMAL’ bumper sticker, sorry for starting off with that question. But you see, COMAL isn’t really all that well known in North America yet, and many people just aren’t sure. If you’re one of the un-COMAL-ized, you may be delighted by what you read here. This article answers the What about COMAL and gives some programming examples just to give you a flavour of the language. A complete COMAL programming tutorial is beyond the scope of this article, but we hope to provide that kind of information in future articles.

COMAL (COMmon Algorithmic Language) is a programming language originally developed in Denmark by Borge Christensen, and is currently in widespread use throughout Europe. It is estimated that there are 100,000 COMAL users worldwide. The first version for Commodore machines ran on the PET/CBM, and was a public domain program, distributed in Canada by Commodore. The new C64 COMAL takes advantage of the 64’s graphics and has been expanded from the original PET version. You can get the C-64 COMAL 0.14 system from the COMAL users group USA (see their address at the end of this article) or make a copy from someone who has it. You are encouraged to make copies of the COMAL system disk for friends or club members, as long as no profits are made and you copy the COMAL system disk unchanged.

COMAL has been described as a cross between BASIC and Pascal, with the good points of both languages and the draw backs of neither. COMAL is as easy to use as BASIC, requiring little overhead to perform simple programs, but it has the speed, control structures and parameter-passing capabilities that BASIC lacks. It does have the powerful structures found in Pascal, but is not as restrictive to the programmer and is simple to use. As another bonus, it also contains the “turtle” graphics commands from LOGO. If this article so far sounds like an endorsement of the COMAL programming language, then we hope it does. Read on about the language’s capabilities and you’ll be able to judge for yourself.

There are two official versions of COMAL in widespread use right now. Version 0.14 runs from disk, and will leave your 64 with about 10K of free memory once the language is loaded into memory. (See the Article “Is 10K Enough?” elsewhere in this issue.) The disk version keeps all error messages on disk to save memory, so there is a slight delay before an error message appears. The newest version of COMAL, called 2.00, comes on a cartridge. The cartridge leaves about 30K of memory free for user programs, runs about twice as fast, and error messages are now fetched instantly. The cartridge also includes new features and commands not found in version 0.14. The points presented below will generally refer to both versions, with exclusive 2.00 features noted in the text.

COMAL is a cross between a compiler and interpreter, compiling each program line as it is entered. That means that you’ll be able to edit and run your programs in the same kind of interactive environment that BASIC enjoys, but your programs will run about 5 to 10 times faster. It also means that the compiler looks at each program line right after you press RETURN, so you’re informed of any syntax errors immediately. This prevents dumb errors...
from sneaking into an obscure part of a program that will only be executed, of course, when you're demonstrating it. If you enter a bad line, the computer beeps, gives a VERY descriptive message, and positions the cursor at the point the error occurred. Fixing the error or moving the cursor to another line will cause the error message to go away and leave the screen EXACTLY the way it was before, as if nothing had ever happened. This is good for the ego, since the computer is so willing to forget your errors and reward your successes.

Programming in COMAL

Many of the actual keywords and functions in COMAL are the same as BASIC, so you won't be totally alienated the first time you fire it up. You still get PEEK, POKE, CHR$, INT, and a lot of other common functions. What makes COMAL better than BASIC is the structure of the language itself. The best thing is that you'll never need GOTOs again, and line numbers have no significance outside of editing--HOORAY! You don't have to worry about indenting your control structures properly, either; COMAL does it for you. The structures available are listed below:

```
IF(condition) . . . THEN . . . ELSE . . . ENDIF
WHILE(condition) . . . ENDWHILE
REPEAT . . . UNTIL(condition)
CASE(expression) . . WHEN(condition) . . OTHERWISE . . . ENDCASE
FOR . . . ENDFOR (like FORNEXT in BASIC)
TRAP . . . HANDLER . . . ENDTRAP (error trap - only in COMAL 2.00)
```

The above control structures are what gives COMAL a superior operative environment to BASIC. You never have to use confusing branches to transfer control to different sections of code; just use the control structures to create a conditional loop or perform a series of instructions or procedures based on a condition. Procedures (explained more later) are like super powerful subroutines, and let you break a problem into simple, understandable modules. Any student of modern structured programming techniques will appreciate COMAL's set-up, and anyone used to Commodore BASIC will be amazed at how much simpler it is to program with an up-to-date, powerful language.

For...Next loops and assignment statements look different from BASIC, but if you enter them in BASIC form, COMAL will automatically convert for you! Version 2.00 will also show all keywords in uppercase when you list the program, and user defined procedures, functions and variables in lowercase.

Besides the structures above, there are other major improvements that COMAL has over Commodore BASIC. For one, the use of long variable names, up to 78 characters long. And all characters are recognized, so 'ACCOUNTS_RECEIVABLE' and 'ACCOUNTS_RECEIVED' are two different variable names. (The underscore is a valid variable name character in version 2.00 and is selected with the back-arrow key.) The other important characteristic of COMAL is its use of procedures and functions.

COMAL Procedures and Functions

When you define a procedure, it's like making your own COMAL keyword, since you call that procedure by just using its name, and passing as many parameters as that procedure needs. For example, a COMAL procedure to draw a square of a given size at a certain angle might look like this:

```
PROC square (size,angle)
    setheading(angle)
    FOR i := 1 TO 4 DO
        forward(size)
        right(90)
    ENDFOR
ENDPROC square
```

Now, to draw a square 25 units large at a 45 degree angle, you would just use the command:

```
EXEC square(25,45)
```

The EXEC statement is optional, so the statement could simply be:

```
square(var1 ,var2)
```

Want a nice design? No problem:

```
FOR n := 1 to 50 DO
    square(n*4,n*5)
ENDFOR
```

Once a procedure has been defined, you can use it from direct mode as well as program mode. A procedure definition can be placed anywhere in a program, and will not be executed unless called; it can't be 'fallen into' like BASIC subroutines. By building a program out of procedures, your code suddenly becomes simpler to understand and easier to de-bug. Further more, a procedure can be defined as 'CLOSED', meaning that all variables defined within the procedure are local. With a closed procedure, you can use any variable names you wish, such as '1', without caring whether it's been used elsewhere. And in version 2.00, if you do wish to use a global variable within a procedure you can bring it in via the IMPORT command. And of
Both COMAL versions contain "turtle" commands such as those found in the language LOGO. Turtle commands, combined with the procedure-programmed nature of COMAL, provide a very easy method to draw incredibly complex patterns on the screen. You simply move around a "turtle" (which you can even create your own packages, customizing the language to your own needs; any package currently in use will be saved along with your program) by pointing him in the right direction and moving him a number of units forward or backward. The main turtle commands are:

- RIGHT and LEFT to turn the turtle a specified number of degrees;
- FORWARD and BACKWARD to move the turtle a specified number of units;
- PENCOLOR to select the drawing colour;
- PENUP and PENDOWN to tell the turtle whether or not to draw as it moves;
- amount := round(cash, 2)

Functions, like procedures, may also be declared as CLOSED, and can be used recursively.

**Features of C64 COMAL**

Besides just the standard COMAL commands, version 0.14 and 2.00 have a whole array of commands to handle graphics and sprites. The cartridge version 2.00 is a complete implementation of COMAL-80, the current standard, but also contains extra commands in the way of packages, which can be invoked with the command:

USE packagename

The concept of packages works well, since the standard COMAL Kernel can be kept machine independent, and extra machine-dependent commands--such as those involving sound, graphics and sprites--can be added at will. That way, you only have to bring in what you need, and not use unnecessary processing time and memory. Some of the packages available with the cartridge version are FONT, GRAPHICS, JOYSTICK, LIGHTPEN, SOUND, SPRITES, SYSTEM and TURTLE. Each of these adds many powerful commands to the language, and additional packages can be loaded from disk. You can even create your own packages, customizing the language to your own needs; any package currently in USE will be saved along with your program.

Both COMAL versions contain "turtle" commands such as those found in the language LOGO. Turtle commands, combined with the procedure-oriented nature of COMAL, provide a very easy method to draw incredibly complex patterns on the screen. You simply move around a "turtle" (which appears as a triangle) by pointing him in the right direction and moving him a number of units forward or backward. The main turtle commands are:

- RIGHT and LEFT to turn the turtle a specified number of degrees;
- FORWARD and BACKWARD to move the turtle a specified number of units;
- PENUP and PENDOWN to tell the turtle whether or not to draw as it moves;
- PENCOLOR to select the drawing colour; and
- amount := round(cash, 2)

The COMAL system isn't just a different language for your C-64, it is an entirely new environment, replacing the 64's ROM set completely and turning the computer into a dedicated COMAL machine. The new environment is familiar, but contains features which help when editing. For one thing, the function keys are set up to generate oft-used commands such as LIST, RUN, TEXTSCREEN, SPLITSCREEN, FULLSCREEN, etc. (TEXTSCREEN and FULLSCREEN select either text or hi-res screen displays. The SPLITSCREEN command displays the hi-res graphics screen while setting a

http://petlibrary.tripod.com/comal4.htm
COMAL has hundreds of features not found in BASIC, too many to list in this article. Things like a built-in string search command, no garbage collection delays, a PRINT USING command for formatted output, a ZONE command to set up tab fields, and dozens of little niceties that there isn't space to mention. At this point though, perhaps you have an idea of the scope and power of the COMAL system, and you can see why many who use it turn into big COMAL fans. Like the ones with the bumper stickers.

COMAL Resources

There are quite a few books on COMAL, both texts and reference. There are also disks available from the COMAL users group packed with programs. The disks are under $10.00 each and there are over 2000 programs on 40 disks available by now. The COMAL users group USA publishes the magazine COMAL Today, which is filled with news, programs, and little tidbits about COMAL. A subscription to COMAL Today also gives you discounts on books and club disks. If you’re interested in learning more about COMAL or wish to start using your COMAL system, a list of good references appear at the end of this article. Reviews of all of these books appeared in COMAL Today #7. These publications, the COMAL 0.14 system, or the cartridge are all available from The COMAL users group, USA. Several packages including COMAL, books and programs are also available. For more information, contact:

COMAL USERS GROUP, U.S.A.

COMAL Book List

"COMAL From A to Z"
Borge Christensen - A reference of all COMAL commands; 64 pages

"COMAL Workbook"
Gordon Shigley -
An exercise text for beginners; 69 pages

"COMAL Library of Functions and Procedures"
Kevin Quiggle -
Reference guide for the included disk; 71 pages

"COMAL 2.0 Packages"
Jesse Knight -
How to add your own ML packages to COMAL; 108 pages

"Beginning COMAL"
Borge Christensen -
Informal introduction to COMAL by its creator; 333 pages

"Captain Comal’s Graphics Primer"
Mindy Skelton -
COMAL graphics and sprites for beginners; 84 pages

"Cartridge Graphics and Sound" Captain Comal’s Friends - Tutorial and reference for 2.0 extra package commands; 64 pages

"Commodore 64 graphics with COMAL" Len Lindsay - Complete organized reference for COMAL graphics commands; 170 pages

"Foundations in Computer Studies with COMAL"
John Kelly -
Programming textbook using COMAL; 363 pages

"Captain Comal Gets Organized"
Len Lindsay -
Writing a disk management system in COMAL, disk included; 102 pages

"Structured Programming with COMAL"
Roy Atherton -
How to write structured COMAL programs; 266 pages

"Cartridge Tutorial Binder"
Frank Bason & Leo Hojsbol-Poulson -
A tutorial specifically for the C64 COMAL 2.0 cartridge; 320 pages

"The COMAL handbook"
Len Lindsay -
The COMAL reference source, giving syntax and sample usage of all standard COMAL-80 commands

Listing 1: COMAL program to draw an N-pointed star -note how COMAL indents the control structures.

http://petlibrary.tripod.com/comal4.htm
"STAR" - this is a sample COMAL
// program to draw a star of any
// number of points.
// transactor magazine 1985 -cz
USE graphics
USE turtle
splitscreen
PRINT " " 147", size := 100
LOOP
PRINT " " 19",
INPUT " number of points? ": points
clear xstart := INT(160-size/2)
ystart := INT(1 00-size/2)
movetos(xstart,ystart)
pendown
star(size,points)
ENDLOOP
PROC star(size,points)
/** draw an N-pointed star **
// first calculate the angle to
// turn at each point
CASE (points MOD 4) OF
WHEN 0
angvar := points
WHEN 2
angvar := points/2
OTHERWISE
angvar := points*2
ENDCASE
angle := 1 80-360/angvar

// now draw the star
setheading((1 80-angle)/2)
FOR i := 1 TO points DO
forward(size)
right(angle)
ENDFOR
ENDPROC star