

## BASIC Steps Up to the GS

by Robert M. Ryan

**T**hey're here. . . .” No, not those little beasts lurking inside your television set, but the first of the 16-bit BASICs for the Apple IIGS. Absoft (AC/BASIC) and TML (TML BASIC) have beaten everyone, including Apple, in the race to develop a version of BASIC that lets you take advantage of certain advanced features of the IIGS, such as graphics and sound, that aren't available from Applesoft BASIC.

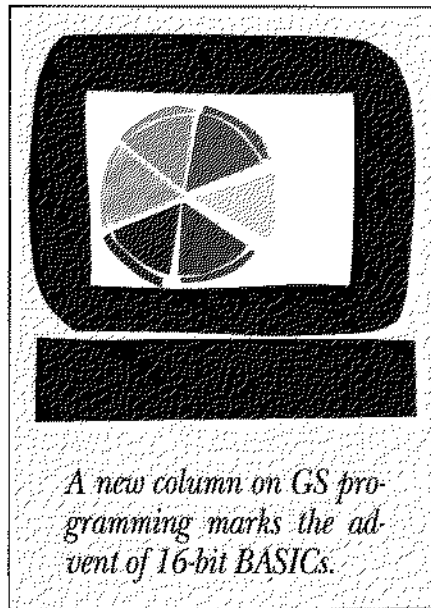
*inCider* has taken this occasion to launch a bimonthly column dedicated to using BASIC to program the GS. It will alternate with Dan Bishop's Applesoft Adviser. Although I hope you'll learn a lot from GS BASICs, I'm not out to produce a straight tutorial. Instead, each installment will focus on an interesting programming project—simulating a stellar system or creating a 3D game that requires stereo glasses, for instance. You'll certainly learn something about programming by studying the listings, but you'll have just as much fun simply entering them and running the programs.

Before getting into the fun stuff, however, one important question remains: What's the difference between TML and AC/BASIC? I've spent the past few weeks fiddling with both; here's what I learned.

### APPLES AND ORANGES

Comparing BASIC compilers is usually a straightforward task. You use each compiler to write code, then compare the speed and size of the resulting programs. Check out special features and the development cycle of each compiler and you usually have a good idea of which one's better.

That's not the case, however, when comparing BASIC compilers for the IIGS. Absoft and TML have taken such radically different approaches that choosing be-



tween Absoft's AC/BASIC and TML BASIC isn't so much a question of performance as it is philosophy.

### BASIC THROUGH THE AGES

BASIC (Beginner's All-purpose Symbolic Instruction Code) was developed at Dartmouth College in the 1960s by John Kemeny and Thomas Kurtz. Their goal was to create a language that would be easy to learn and use. They fulfilled their purpose by inventing a forgiving, English-like syntax that didn't force programmers to learn the details of a computer's innards to write usable programs. Since then, most versions of BASIC, no matter how abbreviated and bastardized, have allowed programmers easy access to the power of their computers.

Absoft's AC/BASIC (2781 Bond Street, Auburn Hills, MI 48057, 313-853-0050, \$125) is a traditional implementation of the language, letting you access the GS' power and special features without having

to understand the complexities of GS hardware and system software. For example, AC/BASIC uses MENU statements to set up pull-down menus and WINDOW statements to open new windows. You don't have to worry about mastering the IIGS ToolBox to use these statements—Absoft has done that for you. You simply have to master Absoft's syntax, an easy enough task.

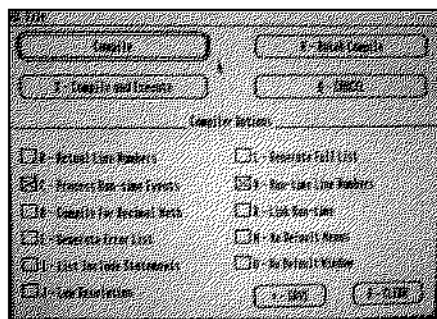
AC/BASIC is the latest in a series of compatible BASIC compilers from Absoft. The company also markets the Microsoft BASIC compiler for the Macintosh and AC/BASIC for the Commodore Amiga. Because Apple produces both the Macintosh and the GS, you shouldn't be surprised to find that compilers for those machines are very compatible. What's more surprising is the degree of compatibility between the Amiga compiler and AC/BASIC for the IIGS. I ported dozens of programs from the Amiga to the GS, most with only minimal modification.

AC/BASIC for the IIGS consists of an editor to enter your program source code and a compiler to translate your source code into machine code. The system comes on a 3½-inch disk and includes a hefty reference manual.

The editor is the weakest part of the AC/BASIC system. If it weren't for its link to the compiler, I would have resorted to another editor or word processor. It's ironic that a system that lets you write programs that take advantage of the GS' event-driven interface comes with an editor that uses nothing from that interface. (If I ever see the word *mode* again I'll scream.)

Once you have a file in the editor you can compile it. Invoking the compiler brings up the Control Window. Here, at a touch of your mouse button, you can set compiler options, which let you create list files of your source code, link your

code to run-time libraries to create stand-alone applications, and suppress the use of standard IIGS menus, among other things. (Suppressing the standard menus is a particularly good idea, because AC/BASIC doesn't let you access them—you can access only the menus you create yourself.) The most important compiler option lets you create programs that respond to run-time events, such as mouse clicks and menu selections. The Control Window also lets you batch-compile a list of files automatically.



The AC compiler—heart of the language.

The AC compiler, the heart of AC/BASIC, understands a rich assortment of statements and functions. It features a large number of variable types and control structures. More importantly, it lets you create colorful, menu-driven programs easily and quickly. It even lets you access some QuickDraw routines (notably those concerned with text styles and graphics primitives) from the ToolBox directly.

Note that calling QuickDraw routines is an option, not a necessity, with AC/BASIC. AC/BASIC tries to give you the power of the ToolBox while insulating you from its complexities. In this, AC/BASIC is very successful.

The downside to this insulation is some inefficiency and loss of precision. High-level statements that access the ToolBox can't be as precise as low-level calls without becoming as complex as those low-level calls. For example, the AC/BASIC WINDOW statement doesn't let you create windows with working close buttons, auto-

### Listing 1. AC/BASIC Mandelbrot generator.

```
'Calculate Mandelbrot Set and Plots
'in 320 X 200 Hires Graphics
'by Bill Kennedy, 1988
'A/C BASIC version by Bob Ryan

'Compile without default menus, with
'default windows, without run-time events.

lft = -2.2
rgt = .6
top = 1.44
bottom = -1.44
stpx = (rgt-lft)/ 320
stpy = (bottom-top)/ 200
xc# = lft

'perform calculation for each on-screen pixel:
FOR i# = 0 TO 319 : 'horizontally
  yc# = top
  FOR j# = 0 TO 199 : 'and vertically
    count# = 0
    x# = xc# : y# = yc#
    xs# = x#^2 : ys# = y#^2
    WHILE count# < 15 AND SQR(xs# + ys#) < 2
      count# = count# + 1
      x# = x#^2 - y#^2
      y# = y#^2 + x#
      xs# = x#^2 : ys# = y#^2
      x# = x# - y# + xc#
      y# = 2 * x# * y# + yc#
    WEND
    'plot the color value = count of the point
    PSET (i#,j#), (15-count#) * 17
    yc# = yc# - stpy : 'next vertical position
  NEXT j#
  xc# = xc# + stpx : 'next horizontal position
NEXT i#

a$ = INPUT$(1)
END
```

### Listing 2. TML BASIC Mandelbrot generator.

```
' Prepare the graphics screen:
library "QuickDraw"
graf init 320
graf on
_clearScreen(0)

' Include the entire Mandelbrot set:

lft = -2.2
rgt = .6
top = 1.44
bottom = -1.44
stpx = (rgt-lft)/ 320
stpy = (bottom-top)/ 200
xc# = lft

'Perform calculation:
For i#0 to 319 ' Pixels horizontal
  yc# = top
  For j#0 to 199 'by vertical
    count# = 0
    x# = xc# : y# = yc#
    xs# = x#^2 : ys# = y#^2
    do 'Reiterate z = z^2 + c
      while count# < 15 and sqrt(xs# + ys#) < 2
        count# = count# + 1
        xs# = x#^2
        ys# = y#^2
        x# = x#
        y# = y#
        x# = x# - y# + xc#
        y# = 2 * x# * y# + yc#
      until
    ' Plot the color value = count:
    SetSolidPenPut(15-count#)
    MoveTo(i#,j#)
    LineTo(i#,j#)
    yc# = yc# + stpy 'Next vertical position
  Next j#
  xc# = xc# + stpx 'Next horizontal position
Next i#

get$ a$ 'Hold screen until a keypress
graf off

END
```

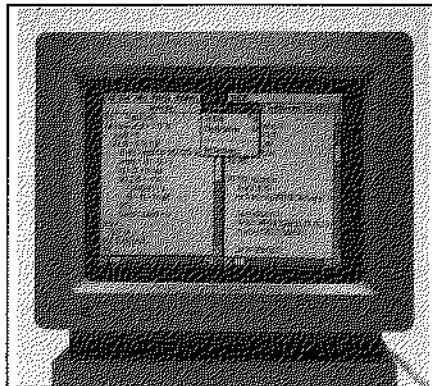
sizing buttons, or scroll bars. AC/BASIC gives you access to much of the power of the ToolBox, but by no means all.

## HEAVY-METAL BASIC

TML BASIC (TML Systems, 8837-B Goodby's Executive Drive, Jacksonville, FL 32217, 904-636-8592, \$125; see Editors' Choice, June 1988, p. 112) takes an entirely different approach toward programming the IIGS. Rather than wrap the ToolBox in a comforting layer of abstraction, TML BASIC requires that you call ToolBox routines directly to perform ToolBox functions. If you don't have a good understanding of ToolBox routines and the organization of GS system software, you won't be able to use TML BASIC to write programs that take advantage of the machine's special features. TML BASIC isn't so much an implementation of the BASIC language as it is a front end for the GS ToolBox.

If you do have a good understanding of the ToolBox, TML BASIC gives you access to all the power of the IIGS. For most people, however, that's a decidedly mixed blessing. The type of precision required of a programmer using TML BASIC is foreign to most Applesoft programmers. BASIC, after all, is supposed to be easy to learn and easy to use. From my experience, TML BASIC is no more accessible than C.

On the up side, TML BASIC features a wonderful editor and a fast compiler.



The TML interface: Menus make it easy.

Unlike the Absoft editor, the TML editor is a mouse- and menu-driven beauty that makes entering and correcting source code a joy. And, although the manual explains the syntax of TML BASIC adequately, it supplies only an introduction to the complexities of ToolBox programming. Although the Absoft manual is no prize, at least it provides all the information you need to begin programming the IIGS.

Absoft's language is a 16-bit version of BASIC that's true to the roots of the language. And although less powerful than TML BASIC, AC/BASIC right now is much more accessible to programmers familiar with Applesoft. Programs appearing in this column will be written

first in AC/BASIC; portions of a program that would be different in TML will be shown in that dialect, too.

## MANDELBROT MANIA

Let's get started using whichever GS BASIC you prefer. The graphics advantage of the GS over earlier Apple II models is best demonstrated perhaps when you call upon the machine to display mathematically generated pictures. *Fractal* mathematics is the current craze—its calculations and graphics representations can exercise your GS to its limits. The accompanying **Listings** use the now-famous *Mandelbrot equation* to do just that.

We'll discuss fractal mathematics in more detail in a later column. For now,

compare the AC and TML BASIC **Listings** to get a better feel for their similarities and differences.

If you run the Mandelbrot program, be prepared to wait a while for a complete picture to emerge. In fact, it's best to run it overnight with all but your GS' power off. That way you'll have a surprise waiting for you with your morning coffee.

Until next time, send your questions and comments to GS BASICs, *inCider*, 80 Elm Street, Peterborough, NH 03458. ■

---

*Bob Ryan is technical editor at AmigaWorld magazine (IDG Communications/Peterborough) and is a contributing editor at inCider. Write to him c/o inCider, 80 Elm Street, Peterborough, NH 03458.*